

TUGAS AKHIR - KI141502

PENERAPAN POLA PERANCANGAN DAN PENGUKURAN KUALITAS PADA SISTEM INFORMASI AKADEMIK: STUDI KASUS MODUL PEMBELAJARAN

PRASETYA GILANG NUSWANTARA
NRP 5113 100 104

Dosen Pembimbing
Dr. Ir. Siti Rochimah, MT
Rizky Januar Akbar, S. Kom., M. Eng.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017



TUGAS AKHIR – KI141502

**PENERAPAN POLA PERANCANGAN DAN
PENGUKURAN KUALITAS PADA SISTEM
INFORMASI AKADEMIK: STUDI KASUS
MODUL PEMBELAJARAN**

PRASETYA GILANG NUSWANTARA
NRP 5113 100 104

Dosen Pembimbing
Dr. Ir. Siti Rochimah, MT
Rizky Januar Akbar, S.Kom., M.Eng.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

[Halaman ini sengaja dikosongkan]



FINAL PROJECT – KI141502

IMPLEMENTATION OF DESIGN PATTERN AND QUALITY MEASUREMENT ON ACADEMIC INFORMATION SYSTEM ON CASE STUDY: LEARNING MODULE

PRASETYA GILANG NUSWANTARA
NRP 5113 100 104

Advisor
Dr. Ir. Siti Rochimah, MT
Rizky Januar Akbar, S.Kom., M.Eng.

INFORMATICS DEPARTMENT
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2017

[Halaman ini sengaja dikosongkan]

**PENERAPAN POLA PERANCANGAN DAN
PENGUKURAN KUALITAS PADA SISTEM INFORMASI
AKADEMIK: STUDI KASUS MODUL PEMBELAJARAN**

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Rekayasa Perangkat Lunak
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :

PRASETYA GILANG NUSWATARA

NRP : 5113 100 104

Disetujui oleh Dosen Pembimbing Tugas Akhir :

Dr. Ir. SITI ROCHIMAH, M.T.

NIP: 196810021994032001



Rizky Januar Akbar, S.Kom., M.Eng.

NIP: 198701032014041001

DEPARTEMEN
TEKNIK INFORMATIKA

(pembimbing 2)

**SURABAYA
JUNI 2017**

[Halaman ini sengaja dikosongkan]

PENERAPAN POLA PERANCANGAN DAN PENGUKURAN KUALITAS PADA SISTEM INFORMASI AKADEMIK: STUDI KASUS MODUL PEMBELAJARAN

Nama Mahasiswa : PRASETYA GILANG NUSWANTARA
NRP : 5113 100 104
Jurusan : Teknik Informatika ITS
Dosen Pembimbing I : Dr. Ir. Siti Rochimah, MT
Dosen Pembimbing II : Rizky Januar Akbar, S.Kom., M.Eng.

Abstrak

Sistem Infomasi Akademik adalah sistem yang dibuat untuk membantu proses bisnis kegiatan akademik. Dalam penerapan sistem informasi akademik, kualitas perlu diperhatikan. Kualitas mempengaruhi performa dari sistem informasi akademik.

Salah satu cara untuk menghasilkan kualitas Sistem Informasi Akademi yang baik adalah dengan membuat rancangan perangkat lunak yang baik dan benar dengan menggunakan pola perancangan. Namun penerapan pola perancangan pada sistem informasi akademik masih dalam tahap penerapan langsung tanpa ada analisis penerapan dan pemilihan pola perancangan dengan kualitas perangkat lunak. Selain itu, kualitas yang dimaksud juga belum jelas.

Untuk memberikan analisis dan rekomendasi atas masalah kualitas dengan penerapan pola perancangan, pada tugas akhir ini akan dibahas analisa pengaruh pola perancangan terhadap kualitas sistem informasi akademik. Untuk standar kualitas yang digunakan adalah ISO 25010 dan ISO 25023 dengan karakteristik kualitas Maintainability. Pola perancangan yang akan diterapkan adalah kelompok Domain Logic.

Analisis penerapan pola perancangan menghasilkan kesimpulan bahwa setiap penerapan pola perancangan pada domain logic akan menghasilkan kualitas maintainability yang berbeda. Dari empat pola perancangan pada domain logic, pola perancangan transaction script mendapatkan nilai sub-kualitas modularity terbaik dengan nilai 0,813. Pola perancangan service

layer mendapatkan nilai sub-kualitas reusability terbaik dengan nilai 0,693 sedangkan untuk sub-kualitas analyzability dan testability tidak didapatkan pola perancangan terbaik karena kedua sub-kualitas tersebut tidak dipengaruhi pola perancangan domain logic.

Kata kunci: sistem informasi akademik, kualitas perangkat lunak, pola perancangan

IMPLEMENTATIONS OF DESIGN PATTERN AND QUALITY MEASUREMENT ON ACADEMIC INFORMATION SYSTEM WITH CASE STUDY: LEARNING MODULE

Name : PRASETYA GILANG NUSWANTARA
NRP : 5113 100 104
Major : Informatics Engineering Department – ITS
Supervisor I : Dr. Ir. Siti Rochimah, MT.
Supervisor II : Rizky Januar Akbar, S.Kom., M.Eng.

Abstract

Academic Information System is system who created to help business process of academic activities. Every Academic Information System must have good quality because quality influence sistem performance.

To create better quality of academic information system, we can create good system design with implementing design pattern. But, nowadays there is no reference to implementation and choosing design pattern for develop Academic Information System. The meaning from quality also can't be descript well.

In this experiment, we will give analysis and recommendation to implementing and choosing design pattern to get better quality of Academic Information System. Quality standarts who will be used are ISO25010 and ISO25023 in Maintainability Characteristic. Design pattern who will be implement is Domain Logic Group.

Analysis of implementing design pattern give conclusion than choose different domain logic design pattern will give different maintainability quality. Transaction script gets 0.813 poin on modularity poin and make it the best of modularity sub-quality. Service layer gets biggest poin with 0.693 poin on reusability sub-quality. Analyzability and testability do not have best domain logic pattern because both of them not affected by domain logic pattern.

Keywords: *Academic Information System, Software Quality, Design Pattern*

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillahirabbil'alamin, segala puji bagi Allah SWT,yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “PENERAPAN POLA PERANCANGAN DAN PENGUKURAN KUALITAS PADA SISTEM INFORMASI AKADEMIK: STUDI KASUS MODUL PEMBELAJARAN”.

Pengerjaan Tugas Akhir ini merupakan suatu kesempatan yang sangat baik bagi penulis. Dengan pengerjaan tugas akhir ini penulis bisa mendapatkan ilmu lebih serta memanfaatkan semua ilmu yang telah didapatkan pada saat berkuliah di Teknik Informatika FTIf ITS ini.

Selesaiannya Tugas Akhir ini tidak lepas dari bantuan dan dukungan beberapa pihak. Sehingga pada kesempatan ini penulis mengucapkan syukur dan terima kasih kepada:

1. Allah SWT dan Nabi Muhammad SAW.
2. Orang tua, Kakak, Adik serta Saudara-saudara yang selalu mendoakan dan mendukung penulis.
3. Ibu Dr.Ir. Siti Rochimah, MT selaku pembimbing I yang selalu memberikan arahan, motivasi dan bantuan sekaligus bimbingan kepada penulis selama pengerjaan Tugas Akhir.
4. Bapak Rizky Januar Akbar, S.Kom, M.Eng selaku pembimbing II yang juga telah sangat membantu, dan membimbing saat pengerjaan Tugas Akhir ini.
5. Mas Agung, Mas Galih, Mas Rahman, Mas Tommy, Mas Bustan dan Mbak Manda sebagai referensi kita dalam mengerjakan Tugas Akhir.
6. Bapak dan Ibu Dosen Karyawan Teknik Informatika FTIf ITS yang telah memberikan ilmunya.

7. Kelompok TA dan Thesis SIA Ibnu, Afif, Mas Khalid dan Mas Baskara yang sudah susah payah menjalankan Tugas Akhir dan Thesis dengan topik SIA ini.
8. Kekasih, yang selalu mendoakan, menghibur dan memberikan ketenangan disaat penulis berada pada titik terberat menjalankan Tugas Akhir ini.
9. “Kontrakan WC Outdoor”, RPL beserta Admin-adminnya yang selalu menjadi naungan dan lingkungan yang sangat mendukung penulis untuk menyelesaikan Tugas Akhir ini.
10. PH HMTC Optimasi, Departemen Riset dan Teknologi HMTC Optimasi dan Trainer Navigator yang menjadi tempat bagi penulis untuk mengistirahatkan pikiran saat mengalami beban mental pengerjaan Tugas Akhir ini.
11. Teman-teman Artcak Teknologi yang juga memberikan pengalaman dan pengetahuan baru serta menjadi lingkungan yang baik bagi penulis di tahun-tahun terakhirnya kuliah.
12. Teman-teman angkatan 2013 yang telah membantu, berbagi ilmu, menjaga kebersamaan, dan memberi motivasi kepada penulis, kakak-kakak angkatan 2012, 2011, serta adik-adik angkatan 2014 dan 2015 yang membuat penulis untuk selalu belajar.
13. Serta semua pihak yang telah turut membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Sehingga dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depannya.

Surabaya, Juni 2017

DAFTAR ISI

Abstrak	vii
Abstract.....	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL	xix
DAFTAR KODE SUMBER	xxv
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah	2
1.4. Tujuan	3
1.5. Manfaat	3
1.6. Metodologi Pembuatan Tugas Akhir	3
1.7. Sistematika Penulisan	5
BAB II DASAR TEORI.....	7
2.1. Sistem informasi akademik	7
2.2. Kualitas Perangkat Lunak	7
2.3. Software Product Quality Requirements and Evaluation (SQuaRE).....	8
2.4. Uji Coba Perangkat Lunak.....	10
2.5. Pola Perancangan	11
2.6. Pola Perancangan Aplikasi <i>Enterprise</i>	12

2.7.	Pola Perancangan <i>Domain Logic</i>	14
BAB III ANALISIS DAN PERANCANGAN SISTEM.....		15
3.1.	Analisis	15
3.1.1.	Uji Coba dan Perbaikan Program Versi Replikasi 19	
3.1.2.	Deskripsi Sistem Saat Ini	19
3.1.3.	Analisis Matrix Kualitas	25
3.2.	Perancangan	37
3.2.1.	Perancangan Transaction Script.....	38
3.2.2.	Perancangan <i>Domain Model</i>	39
3.2.3.	Perancangan <i>Table Module</i>	41
3.2.4.	Perbandingan Pengelompokan Domain Logic....	42
BAB IV IMPLEMENTASI SISTEM		45
4.1.	Lingkungan Pengembangan Sistem	45
4.2.	Penerapan Paket Controller, Validator dan Data	45
4.3.	Penerapan Pola Perancangan <i>Transaction Script</i>	46
4.4.	Penerapan Pola Perancangan <i>Domain Model</i>	51
4.5.	Penerapan Pola Perancangan <i>Table Module</i>	54
4.6.	Proses Deployment Program.....	62
4.7.	Perbandingan Hasil Penerapan Pola Perancangan	63
BAB V PENGUJIAN DAN EVALUASI		69
5.1.	Pengujian Fitur-fitur pada Sistem Baru	69
5.2.	Uji Kualitas Pola Perancangan.....	72
5.2.1.	Kelompok Uji.....	72
5.2.2.	Uji Kualitas Pola Perancangan <i>Service Layer</i>	88

5.2.3.	Uji Kualitas Pola Perancangan <i>Transaction Script</i>	89
5.2.4.	Uji Kualitas Pola Perancangan <i>Domain Model</i> ...	90
5.2.5.	Uji Kualitas Pola Perancangan <i>Table Module</i>	91
5.3.	Perbandingan Hasil Kualitas Sistem	92
5.3.1.	Sub-Kualitas Modularity	95
5.3.2.	Kualitas Reusability	97
5.3.3.	Kualitas Analysability	100
5.3.4.	Kualitas Testability	101
BAB VI KESIMPULAN DAN SARAN		103
6.1.	Kesimpulan	103
6.2.	Saran	104
DAFTAR PUSTAKA		105
LAMPIRAN A		107
LAMPIRAN B		111
LAMPIRAN C		143
LAMPIRAN D		146
BIODATA PENULIS		164

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 4.1. Parameter Kualitas berdasarkan ISO 25010	9
Gambar 5.1. Alur Kerja Tugas Akhir.....	17
Gambar 5.2. Diagram Paket sistem informasi akademik	20
Gambar 5.3. Diagram Paket Modul Pembelajaran	21
Gambar 5.4.Diagram Paket Validator	22
Gambar 5.5.Diagram Kelas PembRepository	23
Gambar 5.6.Diagram Kelas BeritaAcaraController.....	24
Gambar 5.7.Diagram Kelas ThnAjaranService	24
Gambar 5.8. Diagram Kelas Pola Perancangan Service	25
Gambar 5.9.Rancangan Diagram Kelas Pola Perancangan Transaction Script.....	38
Gambar 5.10.Rancangan Diagram Kelas Pola Perancangan Domain Model.....	39
Gambar 5.11.Rancangan Diagram Kelas Pola Perancangan Table Module.....	41
Gambar 7.1.Grafik Hasil Pengujian Kualitas Maintainability	95
Gambar 7.2.Grafik Hasil Pengujian Sub-Kualitas Modularity	95
Gambar 7.3.Grafik Hasil Pengujian Poin Coupling of Component Conformance	96
Gambar 7.4.Grafik Hasil Pengujian Poin Cyclomatic Complecity	96
Gambar 7.5.Grafik Hasil Pengukuran Kualitas Reusability	98
Gambar 7.6.Grafik Hasil Pengukuran Poin Reusability of Assets.....	98
Gambar 7.7.Grafik Hasil Pengukuran Poin Conformance to Coding Rule	99
Gambar 7.8.Grafik Hasil Pengukuran Sub-Kualitas Analysability.....	100
Gambar 7.9.Grafik Hasil Pengukuran Sub-Kualitas Testability	101

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 4.1. Parameter Kualitas Berdasarkan ISO 25023	9
Tabel 4.2. Jenis-jenis pola perancangan aplikasi <i>enterprise</i>..	12
Tabel 5.1.Detail Pengujian Component Coupling.....	27
Tabel 5.2.Detail Pengujian Cyclomatic complexity	28
Tabel 5.3.Detail Pengujian Reusability of Assets	29
Tabel 5.4. Detail Pengujian Conformance to coding rule.....	30
Tabel 5.5.Detail Pengujian System log complateness conformance.....	32
Tabel 5.6.Detail Pengujian Efektifitas dan Diagnosa Fungsi	32
Tabel 5.7.Detail Pengujian Diagnosa Kecukupan Fungsi	33
Tabel 5.8.Detail Pengujian Test function complateness conformance	34
Tabel 5.9.Detail Pengujian Autonomus testability.....	35
Tabel 5.10.Detail Pengujian Kemampuan Ulang Uji Coba...	36
Tabel 5.11. Perbandingan Kelompok Domain Logic.....	43
Tabel 6.1. Pemetaan Kelas Controller dengan Kelas Transaction Scriptnya	50
Tabel 6.2. Pemetaan Kelas Controller dengan Kelas Domain Model.....	51
Tabel 6.3.Fungsi-fungsi Umum pada Kelas-Kelas Paket Tablemodule	55
Tabel 6.4. Pemetaan Kelas Controller dengan Kelas Table Module	59
Tabel 6.5. Pemetaan Kelas Controller dengan Kelas-kelas pada Empat Pola Perancangan.....	64
Tabel 6.6. Perbandingan Kompleksitas Tiap Pola Perancangan	68
Tabel 7.1.Hasil Pengujian Fungsi-Fungsi pada Penerapan Tiga Pola Perancangan Baru	69
Tabel 7.2. Kelompok Fungsi atai Fitur Program	73
Tabel 7.3.Daftar Kelompok Interface	73
Tabel 7.4. Daftar Kelompok Implementasi	80

Tabel 7.5.Daftar Kelompok Uji Coba	87
Tabel 7.6.Ringkasa Hasil Pengukuran Kualitas pada <i>Service Layer</i>	89
Tabel 7.7.Ringkasan Hasil Pengukuran Kualitas pada <i>Transaction Script</i>	90
Tabel 7.8.Ringkasan Hasil Pengukuran Kualitas pada <i>Domain Model</i>	91
Tabel 7.9.Ringkasan Hasil Pengukuran Kualitas Test function complateness conformance pada <i>Table Module</i>	92
Tabel 7.10. Perbandingan Kualitas pada Penerapan Pola Perancangan	93
Tabel 7.11.Ringkasan Perbandingan Kualitas Pola Perancangan	94
Tabel B.1. Detail Pengujian Fungsi Akses Menu	111
Tabel B.2.Detail Pengujian Fungsi Tambah Tahun Ajaran	111
Tabel B.3. Detail Pengujian FungsiSunting Tahun Ajaran	112
Tabel B.4. Detail Pengujian Fungsi Hapus Tahun Ajaran..	112
Tabel B.5. Detail Pengujian Fungsi Tambah Semester	113
Tabel B.6.Detail Pengujian Fungsi Sungtin Semester	114
Tabel B.7.Detail Pengujian Fungsi Hapus Semester.....	114
Tabel B.8.Detail Pegujian Fungsi Tambah Batas Pengambailan SKS.....	115
Tabel B.9. Detail Pengujian Fungsi Sunting Batas Pengambilan SKS.....	116
Tabel B.10.Detail Pengujian FUNgsi Hapus Batas Pengembilang SKS.....	117
Tabel B.11.Detail Pengujian Fungsi Tambah Status Absensi	117
Tabel B.12.Detail Pengujian FUNgsi Sunting Status Absesnis	118
Tabel B.13.Detail Pengujian Fungsi Hapus Status Absensi	119
Tabel B.14.Detail Pengujian Fungsi Tambah Periode.....	119
Tabel B.15.Detail Pengujian Fungsi Sunting Periode.....	120
Tabel B.16.Detail Pengujian Fungsi Hapus Periode	121

Tabel B.17.Detail Pengujian Fungsi Tambah Pembelajaran	121
Tabel B.18.Detail Pengujian Fungsi Sunting Pembelajaran	122
Tabel B.19.Detail Pengujian Fungsi Hapus Pembelajaran	123
Tabel B.20.Detail Pengujian Fungsi Tambah Pengajar	124
Tabel B.21.Detail Pengujian Fungsi Hapus Pengajar	124
Tabel B.22.Detail Pengujian Fungsi Tambah Peserta	125
Tabel B.23.Detail Pengujian Fungsi Tambah Peserta dari Rombel	125
Tabel B.24.Detail Pengujian Fungsi Hapus Peserta	126
Tabel B.25.Detail Pengujian Fungsi Tambah Rombongan Belajar	127
Tabel B.26.Detail Pengujian Fungsi Sunting Rombongan Belajar	127
Tabel B.27.Detail Pengujian Fungsi Hapus Rombel	128
Tabel B.28.Detail Pengujian Fungsi Tambah Anggota	129
Tabel B.29.Detail Pengujian Fungsi Hapus Anggota	129
Tabel B.30.Detail Pengujian Fungsi Tambah Anak Wali	130
Tabel B.31.Detail Pengujian Fungsi Absensi Peserta Didik	131
Tabel B.32.Detail Pengujian Fungsi Absensi Pendidik	132
Tabel B.33.Detail Pengujian Fungsi Mengambil Pembelajaran	133
Tabel B.34.Detail Pengujian Fungsi Menyusun KRS	133
Tabel B.35.Detail Pengujian Fungsi Menambah Pembelajaran	134
Tabel B.36.Detail Pengujian Fungsi Rekap Absensi Peserta Didik	135
Tabel B.37.Detail Pengujian Fungsi Rekap Absensi Pendidik	135
Tabel B.38.Detail Pengujian Fungsi Rekap Berita Acara	136
Tabel B.39.Detail Pengujian Fungsi Persetujuan KRS	136
Tabel B.40.Detail Pengujian Fungsi Membatalkan Persetujuan	137
Tabel B.41.Detail Pengujian Fungsi Berita Acara	138

Tabel B.42.Detail Pengujian Fungsi Pembayaran Peserta Didik	138
Tabel B.43.Detail Pengujian Fungsi Sungting Berita Acara	139
Tabel B.44.Detail Pengujian Fungsi Pengambilan Mata Kuliah ITS	140
Tabel B.45.Detail Pengujian Fungsi Pengambilan Mata Kuliah PENS	140
Tabel B.46.Detail Pengujian Fungsi Pengambilan Matakuliah UPN	141
Tabel B.47.Detail Pengujian Fungsi Perubahan Kebijakan	141
Tabel C.1.Detail Perbaikan Fungsi Akses Menu	143
Tabel C.2. Detail Perbaikan Fungsi Tambah Periode Baru	143
Tabel C.3.Detail Perbaikan Fungsi Edit Periode	144
Tabel C.4.Detail Perbaikan Fungsi Tambah Berita Acara .	144
Tabel C.5.Detail Perbaikan Fungsi Edit Berita Acara	145
Tabel D.1.Hasil Pengukuran Kualitas pada <i>Service Layer</i>.	146
Tabel D.2.Hasil Pengukuran Kualitas Cyclomatic Complexity pada <i>Service Layer</i>.....	146
Tabel D.3.Hasil Pengukuran Kualitas Reusability of Assets pada <i>Service Layer</i>.....	146
Tabel D.4.Hasil Pengukuran Kualitas Conformance to Coding Rule pada <i>Service Layer</i>.....	147
Tabel D.5.Hasil Pengukuran Kualitas System Log Complateness Conformance pada <i>Service Layer</i>.....	147
Tabel D.6.Hasil Pengukuran Kualitas Diagnosis Function Effectiveness pada <i>Service Layer</i>	148
Tabel D.7.Hasil Pengukuran Kualitas Diagnosis Function Suffuance Conformance pada <i>Service Layer</i>	148
Tabel D.8.Hasil Pengukuran Kualitas Test function complateness conformance pada <i>Service Layer</i>	149
Tabel D.9.Hasil Pengukuran Kualitas Autonomus testability pada <i>Service Layer</i>.....	149
Tabel D.10.Hasil Pengukuran Kualitas Kemampuan Restart pada <i>Service Layer</i>.....	149

Tabel D.11.Hasil Pengukuran Kualitas pada Coupling of component conformance <i>Transaction Script</i>	150
Tabel D.12.Hasil Pengukuran Kualitas Cyclomatic Complexity pada <i>Transaction Script</i>	150
Tabel D.13.Hasil Pengukuran Kualitas Reusability of Assets pada <i>Transaction Script</i>	151
Tabel D.14.Hasil Pengukuran Kualitas Conformance to Coding Rule pada <i>Transaction Script</i>	151
Tabel D.15.Hasil Pengukuran Kualitas System Log Completeness Conformance pada <i>Transaction Script</i>	152
Tabel D.16.Hasil Pengukuran Kualitas Diagnosis Function Effectiveness pada <i>Transaction Script</i>	152
Tabel D.17.Hasil Pengukuran Kualitas Diagnosis Function Sufficiency Conformance pada <i>Transaction Script</i>	152
Tabel D.18.Hasil Pengukuran Kualitas Test function completeness conformance pada <i>Transaction Script</i>	153
Tabel D.19.Hasil Pengukuran Kualitas Autonomous testability pada <i>Transaction Script</i>	153
Tabel D.20.Hasil Pengukuran Kualitas Kemampuan Restart pada <i>Transaction Script</i>	154
Tabel D.21.Hasil Pengukuran Kualitas Coupling of component conformance pada <i>Domain Model</i>	154
Tabel D.22.Hasil Pengukuran Kualitas Cyclomatic Complexity pada <i>Domain Model</i>	155
Tabel D.23.Hasil Pengukuran Kualitas Reusability of Assets pada <i>Domain Model</i>	155
Tabel D.24.Hasil Pengukuran Kualitas Conformance to Coding Rule pada <i>Domain Model</i>	155
Tabel D.25.Hasil Pengukuran Kualitas System Log Completeness Conformance pada <i>Domain Model</i>	156
Tabel D.26.Hasil Pengukuran Kualitas Diagnosis Function Effectiveness pada <i>Domain Model</i>	156
Tabel D.27.Hasil Pengukuran Kualitas Diagnosis Function Sufficiency Conformance pada <i>Domain Model</i>	157

Tabel D.28. Hasil Pengukuran Kualitas Test function complateness conformance pada <i>Domain Model</i>.....	157
Tabel D.29. Hasil Pengukuran Kualitas Autonomus testability pada <i>Domain Model</i>	158
Tabel D.30. Hasil Pengukuran Kualitas Kemampuan Restart pada <i>Domain Model</i>	158
Tabel D.31. Hasil Pengukuran Kualitas Coupling of component conformance pada <i>Table Module</i>	158
Tabel D.32. Hasil Pengukuran Kualitas cyclomatic complexity pada <i>Table Module</i>	159
Tabel D.33. Hasil Pengukuran Kualitas reusability of assets pada <i>Table Module</i>	159
Tabel D.34. Hasil Pengukuran Kualitas conformance to coding rule pada <i>Table Module</i>	160
Tabel D.35. Hasil Pengukuran Kualitas system log complateness conformance pada <i>Table Module</i>	160
Tabel D.36. Hasil Pengukuran Kualitas diagnosis function effectiveness pada <i>Table Module</i>	161
Tabel D.37. Hasil Pengukuran Kualitas diagnosis function suffiance conformance pada <i>Table Module</i>	161
Tabel D.38. Hasil Pengukuran Kualitas Test function complateness conformance pada <i>Table Module</i>	161
Tabel D.39. Hasil Pengukuran Kualitas Autonomus testability pada <i>Table Module</i>	162
Tabel D.40. Hasil Pengukuran Kualitas Kemampuan Restart pada <i>Table Module</i>	162

DAFTAR KODE SUMBER

Kode Sumber 6.1.Fungsi Simpan pada Kelas PdTransaction	
.....	49
Kode Sumber 6.2.Fungsi pdSave pada Kelas PdTransaction	
.....	50
Kode Sumber 6.3.Implementasi Salah Satu Kelas Pada <i>Table Module</i>	
.....	59

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Bab ini membahas garis besar penyusunan tugas akhir yang meliputi latar belakang, tujuan pembuatan, rumusan dan batasan permasalahan, metodologi penyusunan tugas akhir, dan sistematika penulisan.

1.1. Latar Belakang

Kemajuan teknologi informasi saat ini membuat seseorang tidak perlu melakukan tatap muka dengan orang lain dalam melakukan transaksi tertentu. Begitu pula dalam kegiatan akademik. Dalam menerapkan teknologi informasi pada kegiatan akademik, salah satu penerapannya adalah sistem informasi akademik. Sistem informasi akademik adalah sistem yang digunakan suatu institusi pendidikan untuk menjalankan seluruh proses bisnis utama dari kegiatan pendidikan, seperti penilaian dan absensi. Dalam penerapan sistem informasi akademik, kualitas perlu diperhatikan agar sistem memiliki kinerja yang lebih baik. Kualitas akan menentukan apakah sistem tersebut aman, mudah digunakan, dan sesuai dengan semua kebutuhan yang telah ditentukan.

Sebuah sistem seperti sistem informasi akademik yang kualitasnya sudah dianggap baik sering kali tetap terdapat kelemahan yang muncul seperti bug, error atau performa yang masih kurang baik. Karena itu perlu ada perbaikan kualitas perangkat lunak. Salah satu cara mendapatkan kualitas perangkat lunak adalah dengan membuat rancangan perangkat lunak yang baik dengan menerapkan pola perancangan. Namun saat ini penerapan pola perancangan masih dilakukan langsung tanpa ada analisis terhadap pola perancangan dan kualitas. Selain itu kualitas yang dimaksud sendiri belum memiliki standar.

Pada tugas akhir ini akan membahas analisis pengaruh penerapan dan pemilihan penerapan pola perancangan terhadap kualitas sistem informasi akademik. Pola perancangan yang

digunakan adalah pola perncangan pada kelompok Domain Logic sedangkan standar kualitas yang digunakan adalah ISO25010 dan ISO25023 pada karakteristik kualitas Maintainability.

Hasil yang diharapkan adalah hasil analisis pola perancangan dan pengaruhnya terhadap kualitas sistem informasi akademik. Serta didapatkan referensi atau rekomendasi pemilihan pola perancangan yang tepat.

1.2. Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

- a. Bagaimana cara menerapkan pola perancangan kelompok Domain Logic pada sistem informasi akademik?
- b. Bagaimana cara mengukur kualitas sistem informasi akademik dengan parameter kualitas Maintainability?
- c. Bagaimana cara mengevaluasi hasil pengukuran sistem informasi akademik?
- d. Apakah penerapan pola perancangan berpengaruh terhadap kualitas sistem informasi akademik?

1.3. Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir memiliki beberapa batasan, yakni sebagai berikut.

- a. Standar yang digunakan adalah ISO/IEC 25010 dan ISO/IEC DIS 25023.
- b. sistem informasi akademik ini berbasis web dengan bahasa pemrograman Java dengan kerangka kerja Spring dan basis data PostgreSQL.
- c. sistem informasi akademik yang digunakan adalah sistem informasi akademik versi penelitian (replika sistem informasi akademik), versi ini tidak sama dengan versi yang digunakan pada Institut Teknologi Sepuluh Nopember[1]–[6].
- d. Basis data yang digunakan adalah basis data replika dan tidak akan mengalami perubahan.

- e. Modul yang akan dikembangkan adalah Modul Pembelajaran.

1.4. Tujuan

Tujuan pembuatan tugas akhir ini antara lain:

- a. Mengetahui cara menerapkan pola perancangan kelompok Domain Logic pada sistem informasi akademik.
- b. Mengetahui cara mengukur kualitas sistem informasi akademik dengan parameter kualitas Maintainability.
- c. Mengetahui cara mengevaluasi hasil pengukuran sistem informasi akademik.
- d. Mengetahui pengaruh pola perancangan terhadap kualitas sistem informasi akademik.

1.5. Manfaat

Manfaat dari hasil pembuatan tugas akhir ini antara lain :

- a. Untuk memberikan bukti ilmiah bahwa penerapan pola perancangan dapat mempengaruhi kualitas sistem informasi akademik.
- b. Sebagai referensi untuk pengembangan sistem informasi akademik

1.6. Metodologi Pembuatan Tugas Akhir

1. Penyusunan proposal tugas akhir

Proposal tugas akhir ini berisi tentang deskripsi pendahuluan dari tugas akhir yang akan dibuat. Pendahuluan ini terdiri atas hal yang menjadi latar belakang diajukannya usulan tugas akhir, rumusan masalah yang diangkat, batasan masalah untuk tugas akhir, tujuan dari pembuatan tugas akhir, dan manfaat dari hasil pembuatan tugas akhir. Selain itu dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi pendukung pembuatan tugas akhir. Sub bab metodologi berisi penjelasan mengenai tahapan penyusunan tugas akhir mulai dari penyusunan proposal hingga penyusunan buku tugas akhir. Terdapat pula sub bab jadwal kegiatan yang menjelaskan jadwal pengerjaan tugas akhir.

2. Studi literatur

Pada studi literatur ini, akan dipelajari sejumlah referensi yang diperlukan dalam pembuatan aplikasi yaitu mengenai parameter kualitas perangkat lunak, cara pengukuran kualitas perangkat lunak, ISO 25010, ISO 25023, *Design Pattern*, *Enterprise Application Pattern*.

3. Analisis dan desain perangkat lunak

Analisa dan desain perangkat lunak yang kami kembangkan akan berbasis pada sistem informasi akademik yang telah dikembangkan sebelumnya. Pada tahap ini juga kami melakukan *Reverse Engineering* terhadap *source code* dari sistem informasi akademik yang telah dikembangkan sehingga didapatkan diagram paket dan diagram kelas. Kami juga menentukan parameter-parameter uji yang nantinya akan kami gunakan dalam uji kualitas perangkat lunak.

4. Implementasi perangkat lunak

sistem informasi akademik ini dibangun dengan bahasa pemrograman java dengan *framework* Java Swing. Untuk mengembangkan aplikasi ini kami menggunakan *Integrated Development Environment* (IDE) Eclipse dan PostgreSQL sebagai *Relational Database Management System* (RDBMS).

Tahap implementasi ini dibagi menjadi tiga bagian yaitu uji kualitas program saat ini, penerapan pola perancangan dan uji kualitas program yang telah menerapkan pola perancangan.

5. Pengujian dan evaluasi

Pengujian dilakukan sesuai dengan ketentuan yang telah ada dalam ISO 25010[1] dan ISO 25023[2] terutama pada penilaian internal *source code*. Evaluasi dilakukan dengan cara membandingkan nilai kualitas antara sebelum dan setelah implementasi pattern.

6. Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam

tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat.

1.7. Sistematika Penulisan

Buku Tugas Akhir ini terdiri atas beberapa bab yang tersusun secara sistematis, yaitu sebagai berikut.

1. BAB 1, Pendahuluan, menjelaskan latar belakang, batasan masalah, tujuan dari pembuatan tugas akhir ini serta metodologi yang digunakan selama penyusunan.
2. BAB 2, Dasar Teori, memaparkan hasil studi literatur yang digunakan sebagai dasar untuk menyelesaikan tugas akhir ini, terdiri atas konsep parameter kualitas perangkat lunak, cara pengukuran kualitas perangkat lunak, ISO 25010, ISO 25023, *Design Pattern*, *Enterprise Application Pattern*.
3. BAB 3, Analisa dan desain perangkat lunak yang kami kembangkan akan berbasis pada sistem informasi akademik yang telah dikembangkan sebelumnya. Pada tahap ini juga kami melakukan *Reverse Engineering* terhadap *source code* dari sistem informasi akademik yang telah dikembangkan sehingga didapatkan diagram paket dan diagram kelas. Kami juga menentukan parameter-parameter uji yang nantinya akan kami gunakan dalam uji kualitas perangkat lunak.
4. BAB 4, Implementasi Sistem, sistem informasi akademik ini dibangun dengan bahasa pemrograman java dengan *framework* Java Swing. Untuk mengembangkan aplikasi ini kami menggunakan *Integrated Development Environment* (IDE) Eclipse dan PostgreSQL sebagai *Relational Database Management System* (RDBMS). Tahap implementasi ini dibagi menjadi tiga bagian yaitu uji kualitas program saat ini, penerapan pola perancangan dan uji kualitas program yang telah menerapkan pola perancangan.
5. BAB 5, Pengujian dan Evaluasi, Pengujian dilakukan sesuai dengan ketentuan yang telah ada dalam ISO 25010[1] dan ISO 25023[2] terutama pada penilaian internal *source code*.

Evaluasi dilakukan dengan cara membandingkan nilai kualitas antara sebelum dan setelah implementasi pattern.

6. BAB 6, Kesimpulan dan Saran, berisi tentang kesimpulan yang didapat dari proses pembuatan tugas akhir beserta saran-saran untuk pengembangan selanjutnya.

BAB II DASAR TEORI

Bab ini membahas teori-teori yang mendukung pembuatan tugas akhir. Teori yang mendukung tersebut adalah deskripsi mengenai sistem informasi akademik, kualitas perangkat lunak, Software Product Quality Requirements and Evaluation (SQuaRE), uji coba perangkat lunak, pola perancangan dan pola perancangan perangkat lunak enterprise.

2.1. Sistem informasi akademik

Sistem informasi akademik adalah sistem informasi yang digunakan pada institusi pendidikan. sistem informasi akademik memiliki beberapa fungsionalitas yang dapat membantu menjalankan proses bisnis dari suatu institusi pendidikan. Pada tugas akhir yang akan kami kembangkan, kami menggunakan sistem informasi akademik versi penelitian (replika sistem informasi akademik).

Pada sistem informasi akademik yang telah dikembangkan memiliki 6 modul yaitu modul Framework, Domain, Ekivalensi, Kurikulum, Pembelajaran dan Penilaian. pada pengembangannya sendiri menggunakan bahasa pemrograman java, *framework* Spring dan *database* PostgreSQL[3]–[8].

2.2. Kualitas Perangkat Lunak

Kualitas perangkat lunak adalah pemenuhan terhadap kebutuhan fungsional dan kinerja yang didokumentasikan secara eksplisit, pengembangan standar yang didokumentasikan secara eksplisit dan sifat-sifat implisit yang diharapkan dari sebuah perangkat lunak yang dibangun secara profesional[9]. Sedangkan, perangkat lunak sendiri dikatakan berkualitas apabila memenuhi tiga ketentuan pokok:

1. Memenuhi kebutuhan pemakai yang berarti bahwa jika perangkat lunak tidak dapat memenuhi kebutuhan pengguna

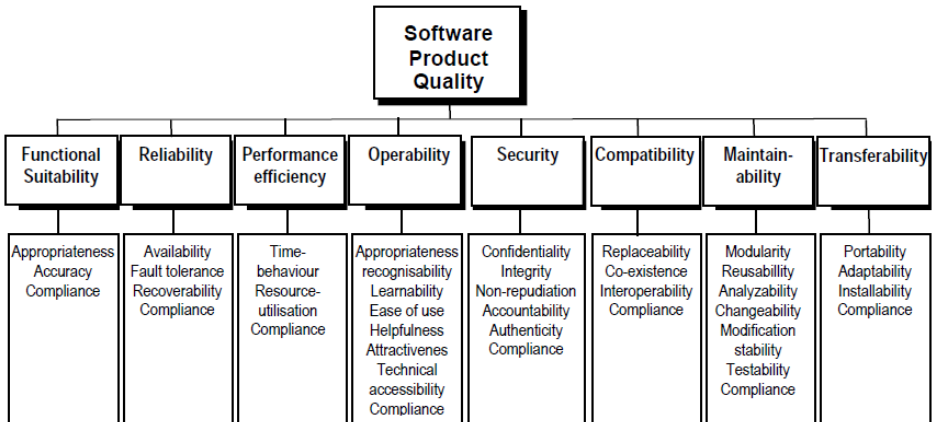
- perangkat lunak tersebut, maka yang bersangkutan dikatakan tidak atau kurang memiliki kualitas.
2. Memenuhi standar pengembangan perangkat lunak , yang berarti bahwa jika cara pengembangan perangkat lunak tidak mengikuti metodologi standar, maka hampir dapat dipastikan bahwa kualitas yang baik akan sulit atau tidak tercapai.
 3. Memenuhi sejumlah kriteria implisit, yang berarti bahwa jika salah satu kriteria implisit tersebut tidak dapat dipenuhi, maka perangkat lunak bersangkutan tidak dapat dikatakan memiliki kualitas yang baik[10].

2.3. Software Product Quality Requirements and Evaluation (SQuaRE)

Quality Model adalah dasar dari sistem evaluasi dari kualitas suatu produk. *Quality Model* menentukan karakteristik kualitas yang akan digunakan dalam mengevaluasi sebuah produk perangkat lunak.

Kualitas sebuah sistem akan menentukan sejauh mana sebuah sistem memenuhi kebutuhan yang telah ditentukan oleh semua pemangku kepentingan, sehingga menghasilkan sebuah nilai. Semua kebutuhan dari para pemangku kepentingan akan direpresentasikan didalam *Quality Model*, yang akan mengkategorikan kualitas produk menjadi karakteristik dan sub-karakteristik[1]

Quality Model sendiri memiliki beberapa standar internasional yaitu ISO/IEC 25010 dan ISO/IEC 25023. pada ISO/IEC 25010 terdapat 8 parameter kualitas yang digambarkan pada gambar 2.1. Sedangkan pada ISO/IEC 25023 parameter kualitas dijelaskan pada tabel 2.1[2].



Gambar 2.1. Parameter Kualitas berdasarkan ISO 25010

Tabel 2.1. Parameter Kualitas Berdasarkan ISO 25023

No	Quality Characteristic	Quality Sub-Characteristic
1	Functional Suitability	Functional Completeness
		Functional Correctness
		Functional appropriateness
2	Performance efficiency	time behaviour
		resource utilization
		capacity
3	compatibility	co-existence
		interoperability
4	Usability	appropriateness recognisability
		Learnability
		operability
		user error protection
		user interface aesthetics

No	Quality Characteristic	Quality Sub-Characteristic
5	reability	accessibility
		maturity
		availability
		avaibility
		fault tolerance
		recoverability
6	security	confidentiality
		integrity
		non-repudiation
		accountability
		authenticity
7	maintainability	modularity
		reusability
		analysability
		midifiability
		testability
8	portability	adaptability
		installability
		replaceability

2.4. Uji Coba Perangkat Lunak

Menurut IEEE Std 610.12 uji coba perangkat lunak memiliki dua pengertian, pengertian pertama adalah proses penggunaan sistem atau komponen dalam kondisi yang tertentu dengan pengamatan dan pencatatan hasil serta membuat evaluasi dari aspek sistem atau komponen. Sedangkan pengertian kedua adalah proses menganalisa perangkat lunak untuk memperoleh

perbedaan antara hasil sistem dan kebutuhan serta melakukan evaluasi terhadap fitur-fiturnya [11].

Berdasarkan konsepnya, uji coba perangkat lunak dibagi menjadi dua jenis yaitu Black Box Testing dan White Box Testing. Uji coba Black Box berfokus pada fungsionalitas dari suatu perangkat lunak baik secara fungsi maupun hasil tanpa memperhatikan proses di dalam sistem. White Box Testing adalah uji coba terhadap proses dalam sistem. Uji coba ini dapat menghasilkan detail mekanisme dan proses di dalam suatu perangkat lunak.

2.5. Pola Perancangan

Pola Perancangan adalah sebuah solusi general yang digunakan pada masalah yang sering terjadi pada pengembangan perangkat lunak. Pola perancangan dikembangkan oleh Gang of Four. Pola perancangan dapat membantu mempermudah pengembangan perangkat lunak[12]. Pola perancangan sendiri dibagi menjadi tiga kelompok yaitu Creational Pattern, Structural Pattern dan Behavioral Pattern.

Creational Pattern adalah pola perancangan yang berhubungan dengan mekanisme pembentukan objek. Tujuan penggunaan pola perancangan ini adalah mengurangi kompleksitas pembuatan objek serta mempermudah mengontrol pembuatan objek. *Creational Pattern* sendiri memiliki enam jenis pola perancangan yaitu *Abstract Factory*, *Builder*, *Factory Method*, *Object Pool*, *Prototype* dan *Singleton*.

Structural Pattern adalah pola perancangan yang mendesain pola hubungan antar entitas. *Structural Pattern* sendiri memiliki 8 jenis pola perancangan yaitu *Adapter*, *Bridge*, *Composite*, *Decorator*, *Facade*, *Flyweight*, *Private Class Data* dan *Proxy*.

Behavioral Pattern adalah pola perancangan yang berhubungan dengan komunikasi antar objek serta realisasi dari objek itu sendiri. Penerapan pola perancangan ini bertujuan untuk meningkatkan *flexibility* dari komunikasi antar objek. *Behavioral Pattern* memiliki 12 jenis pola perancangan yaitu *Chain of*

Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Null Object, Observer, State, Strategy, Template Method dan Visitor.

2.6. Pola Perancangan Aplikasi *Enterprise*

Pola perancangan aplikasi *enterprise* merupakan pola-pola yang sering digunakan untuk mengembangkan aplikasi *enterprise*. Pada bukunya “Pattern of Enterprise Application Architecture”, Martin Fowler memberikan penjelasan tentang bagian-bagian pola perancangan yang dapat digunakan untuk mengembangkan aplikasi *enterprise*[13]. Pola perancangan tersebut ditunjukkan pada tabel 2.2.

Tabel 2.2. Jenis-jenis pola perancangan aplikasi *enterprise*.

No	Tip	Tujuan	Pola Perancangan
1	Domain Logic Pattern	Mengelola Domain Logic	<i>Transaction Script</i>
			<i>Domain Model</i>
			Table Modul
2	Data Source Architecture Pattern	Membentuk hubungan antara Domain Logic dengan <i>database</i>	Table Data Gateway
			Row Data Gateway
			Active Record
			Data Mapper
3	Object Relational Behavioral Pattern	Menentukan pola memanggil dan menyimpan suatu objek ke <i>database</i>	Unit of Work
			identity map
			Lazy Load
4	Object Relational Structural Pattern	Mengorganisasi pemetaan antara objek yang berada pada program/ <i>memory</i> dengan <i>database</i> serta mengorganisasi	Identity Field
			Foreign Key Mapping
			Association Table Mapping
			Dependent Mapping
			Embedded Value

No	Type	Tujuan	Pola Perancangan
		struktur penurunan kelas objek dan pemetaannya ke <i>database</i>	Serialized LOB
			Single Table Inheritance
			Class Table Inheritance
			Concrete Table Inheritance
			Inheritance Mappers
5	Object-Relational Metadata Mapping Patterns	Mengorganisasi pemetaan dalam bentuk <i>metadata</i>	Metadata Mapping
			Query Object
			Repository
6	Web Presentation Patterns	mengorganisasi pemanggilan dan tampilan sebuah program	Model View Controller
			Page Controller
			Front Controller
			Template View
			Transform View
			Two Step View
7	Distribution Patterns	Mengorganisasi proses distribusi objek	Application Controller
			Remote Facade
8	Offline Concurrency Patterns	Mengorganisasi <i>concurenrrency</i> atau beberapa proses yang berjalan bersama-sama serta menggunakan sumber daya yang sama	Data Transfer Object
			Optimistic Offline Lock
			Pessimistic Offline Lock
			Coarse-Grained Lock
9	Session State Patterns	Mengelola penempatan dan penggunaan <i>session</i>	Implicit Lock
			Client Session State
			Server Session State
			Database Session State

2.7. Pola Perancangan *Domain Logic*

Pola perancangan *domain logic* adalah pola perancangan yang diterapkan pada lapisan Domain. Pola perancangan ini bertujuan untuk mengelompokkan business logic pada sistem. Pola perancangan ini memiliki empat jenis pola yaitu *transaction script*, *domain model*, *table module* dan *service layer*.

Pada pola perancangan *transaction script* adalah pola perancangan yang mengelompokkan proses logika pada tiap-tiap transaksi. Pola perancangan ini adalah pola perancangan yang paling sederhana dibanding dengan pola perancangan domain logic lainnya. Pola perancangan domain model adalah pola perancangan pada kelompok *Domain Logic* yang mengelompokkan proses-proses bisnis ke objek-objek individu yang merepresentasikan domain-domain masalah dari sistem. Setiap objek pada *Domain Model* akan berisi gabungan proses dari sebuah kelompok masalah.

Table Module adalah pola perancangan pada kelompok *Domain Logic* yang mengelompokkan proses-proses bisnis berdasarkan table yang telah dibuat. Setiap *Table Module* akan berisi proses-proses yang dibutuhkan lapisan presentasi untuk mengambil data dari table *database*. Pola perancangan terakhir adalah pola perancangan *service layer*. *Service Layer* sendiri adalah pola perancangan yang menghubungkan *data source* dengan proses bisnis ataupun dengan komponen lain seperti tampilan antarmuka sistem.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas tahap analisis permasalahan dan perancangan tugas akhir. Pada bagian awal dibahas mengenai analisis permasalahan yang ingin diselesaikan. Selanjutnya dibahas mengenai perancangan program untuk memberikan gambaran umum mengenai perubahan sistem yang dibuat. Pendekatan yang digunakan dalam perancangan ini adalah pendekatan rancangan berorientasi objek yang direpresentasikan dengan menggunakan diagram UML (*Unified Modelling Language*).

3.1. Analisis

Proses penerapan pola perancangan serta pengukuran kualitas pada sistem informasi akademik memiliki beberapa tahap yang harus dilalui (gambar 3.1). Berikut tahap-tahap yang harus dilakukan untuk menerapkan

a. Rekayasa Balik

Proses rekayasa balik adalah proses menganalisis *source code* sistem informasi akademik dan merubah *source code* yang ada menjadi diagram paket dan diagram kelas guna mendapatkan desain dari sistem informasi akademik yang telah ada. Pada proses ini dibutuhkan program sistem informasi akademik versi replikasi serta dokumen uji cobanya.

Proses rekayasa balik memiliki tiga proses utama yaitu uji coba program, perbaikan program serta analisis struktur program. Proses-proses rekayasa balik akan dijelaskan lebih detail pada bab 3.1.1 dan bab 3.1.2.

b. Analisis Matrix Kualitas

Pada tahap ini kami menentukan parameter kualitas apa saja yang akan kami gunakan dalam uji kualitas. Sumber uji kualitas kami adalah ISO 25010 dan ISO 25023. Pada topik

tugas akhir kami ini berfokus pada parameter uji internal atau *source code*. Pembahasan proses ini ada pada Bab 3.1.3

c. Perancangan Program Baru

Pada tahap ini kami menentukan pola perancangan yang akan diterapkan pada modul pembelajaran. Setelah menentukan pola perancangan kami membuat desain pola perancangan yang direpresentasikan dengan diagram kelas. Perancangan akan dibahas pada bab 3.2

d. Implementasi Program Baru

Pada penerapan pola perancangan kami menerapkan pola-pola perancangan yang telah dirancang ke kode program sehingga terbentuklah program baru yang memiliki proses bisnis sama namun dibuat dengan pola perancangan yang berbeda. Implementasi pola perancangan akan dibahas pada Bab 4.

e. Uji Coba Program Baru

Tujuan dari proses ini adalah pengujian fitur program. Pengujian fitur program dilakukan untuk memastikan fitur-fitur pada program yang telah diubah tidak berubah dari program asli. Pembahasan uji coba akan dibahas pada bab 5.1

f. Uji Kualitas Program

Uji Kualitas Program adalah pengujian kualitas dari masing-masing program. Dalam pengujian kualitas kami menggunakan ISO 25010 dan ISO 25023. Hasil uji kualitas dari tiap-tiap program akan dibandingkan. Proses pengujian ini ada pada Bab 5.2.2 hingga 5.2.5

g. Analisis dan Evaluasi

Pada tahap ini kami akan membandingkan hasil uji kualitas. Setelah mendapatkan hasil perbandingan hasil uji kualitas kami dapat melakukan evaluasi dari hasil implementasi pola perancangan pada sistem informasi akademik

Gambar 3.1. Alur Kerja Tugas Akhir

[Halaman ini sengaja dikosongkan]

3.1.1. Uji Coba dan Perbaikan Program Versi Replikasi

Sistem informasi akademik versi replikasi yang kami kembangkan ini memiliki beberapa kesalahan yang harus diperbaiki terlebih dahulu sebelum dikembangkan lebih jauh lagi. Dalam melakukan uji coba, kami menggunakan menggunakan skenario dari pengembang sebelumnya [3]. Skenario Uji Coba ini juga akan digunakan untuk melakukan uji coba pada sistem yang telah dirubah. Uji Coba sendiri berjalan pada lingkungan

1. Perangkat keras komputer *server*
 - i. Prosesor: Intel® Core™ i3-23100M CPU @ 2.10GHz
 - ii. *Memory*(RAM): 4,00 GB
 - iii. Tipe sistem: 64-bit sistem operasi
2. Perangkat lunak komputer *server*
 - i. Sistem operasi: Windows 8 Professional.
 - ii. Server aplikasi web: Pivotal tc Server Integration.
 - iii. Manajemen basis data: PostgreSQL.
3. Perangkat lunak komputer *client*
 - i. *Web browser*: Google Chrome.

Detail uji coba dapat dilihat pada Lampiran A. Dari hasil uji coba ternyata benar ada fitur yang masih terjadi error. Fitur-fitur tersebut adalah

1. Akses menu (skenario uji PF-000)
2. Tambah periode baru (skenario uji PF-013)
3. Edit periode (skenario uji PF-014)
4. Tambah berita acara (skenario uji PF-041)
5. Edit berita acara (skenario uji PF-043)

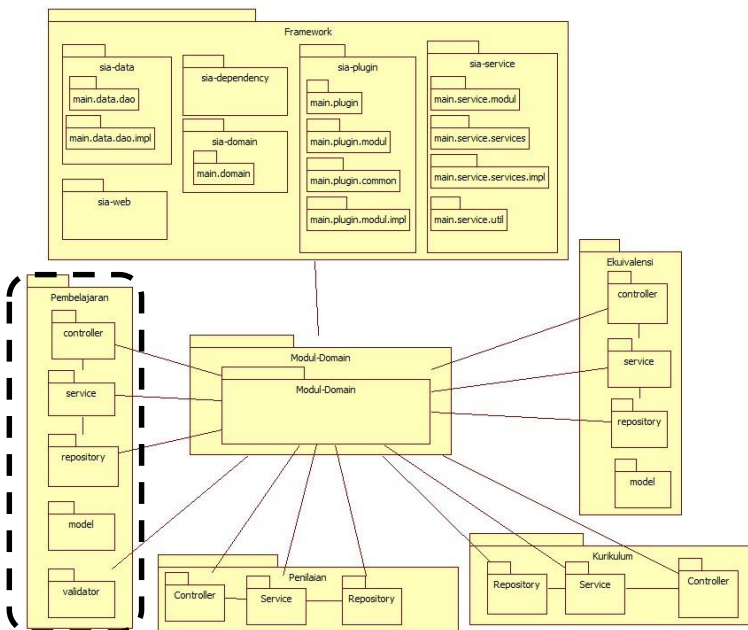
Dari fitur-fitur diatas telah dilakukan perbaikan program (detail pada Lampiran B) sehingga program dapat berjalan dengan baik.

3.1.2. Deskripsi Sistem Saat Ini

Sistem informasi akademik yang digunakan adalah sistem informasi akademik versi penelitian. Sistem Akademik ini terdiri dari 6 modul yaitu modul framework, domain, pembelajaran,

ekuivalensi, kurikulum, dan penilaian. Modul framework adalah modul pusat, Modul domain sebagai penghubung modul lain dengan *database*, dan 4 modul lain adalah modul yang menjalankan fungsi akademik.

sistem informasi akademik ini menggunakan bahasa pemrograman java dengan Spring MVC untuk *web development*. Selain itu arsitektur modul didukung dengan menggunakan Eclipse Virgo dan OSGI Framework. *Web Server* yang digunakan adalah Tomcat.



Gambar 3.2. Diagram Paket sistem informasi akademik

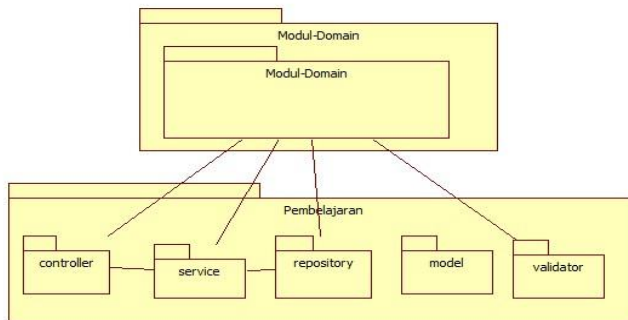
3.1.2.1. Rekayasa Balik

Proses rekayasa balik adalah proses merubah source code menjadi diagram-diagram sehingga kami dapat mengetahui desain dan rancangan dari system ini. Secara garis besar, sistem informasi

akademik yang akan kami kembangkan memiliki Desain seperti pada gambar 3.2.

Pada diagram paket gambar 3.3 menjelaskan bahwa sistem informasi akademik terbagi menjadi 6 modul yaitu modul framework, domain, ekuivalensi, kurikulum, penilaian dan pembelajaran. Topik tugas akhir ini akan berfokus pada modul Pembelajaran. Pada modul pembelajaran ini, pengembang sebelumnya telah menerapkan pola perancangan pada *Domain Logic* yaitu *Service Layer*. Sebagai pembandingan kompleksitas program, berikut kompleksitas program saat ini

- Jumlah Kelas: 147 Kelas
- Total Line of Code: 5522 Line
- Total Method: 726 Method
- Total Cyclomatic: 1427

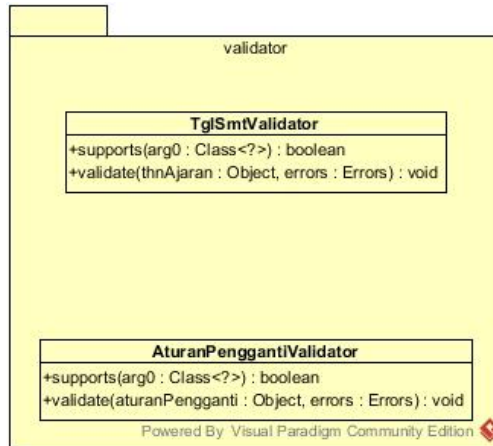


Gambar 3.3. Diagram Paket Modul Pembelajaran

Modul pembelajaran sendiri memiliki 5 paket di dalamnya yaitu:

1. Paket Validator

Paket memiliki fungsi sebagai validasi data masuk dan peraturan-peraturan tentang proses bisnis pembelajaran. Berikut class diagram hasil rekayasa balik dari paket validator



Gambar 3.4. Diagram Paket Validator

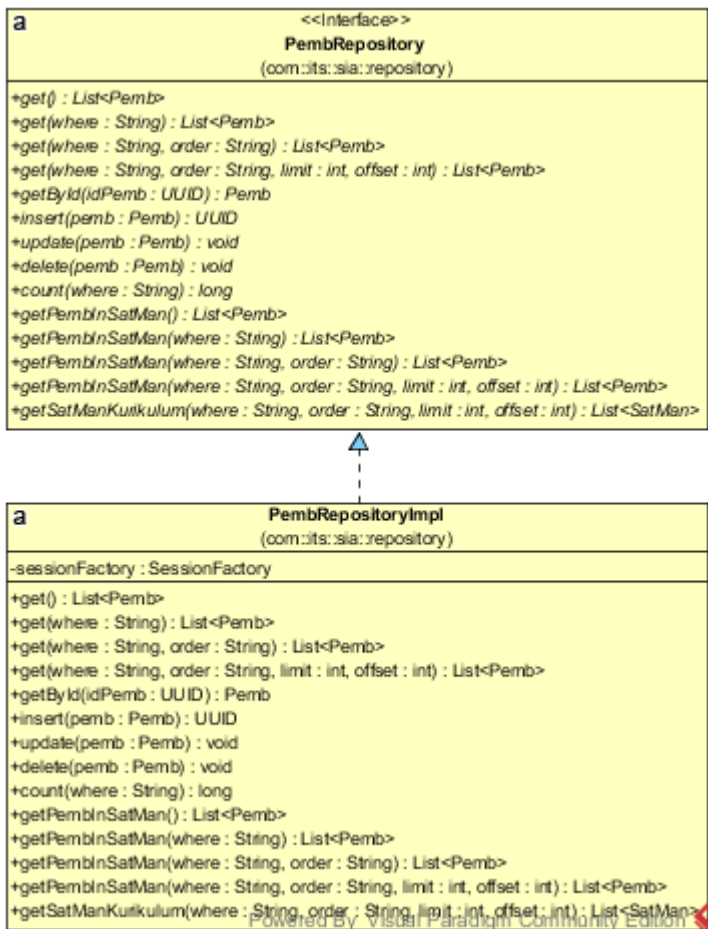
2. Paket Model

Merupakan paket yang berisi entity yang merepresentasikan table, kolom dan relasi didalamnya. Namun paket model ini sudah tidak digunakan karena telah *direfactor* pada pengembangan aplikasi sebelumnya. Model saat ini menggunakan paket modul domain.

Berikut hasil rekayasa balik dari paket model

3. Paket Repository

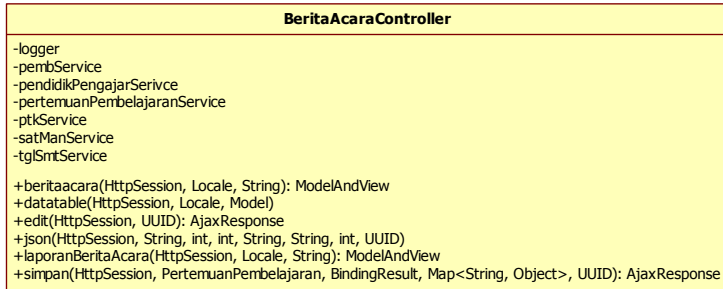
Paket ini merupakan kelas yang digunakan untuk mengambil, memasukan dan mengubah data dalam basis data. Koneksi ke basis data dalam kelas ini ditangani oleh *session factory*. Contoh salah satu diagram kelas pada paket repository dapat dilihat pada gambar 3.5.



Gambar 3.5. Diagram Kelas PembRepository

4. **Paket Controller**
 Paket ini berisi kelas-kelas *controller* yang berfungsi sebagai penghubung layer presentasi dengan layer sumber data. *Controller* juga memiliki fungsi untuk menerima request dan memberikan return berupa data atau halaman

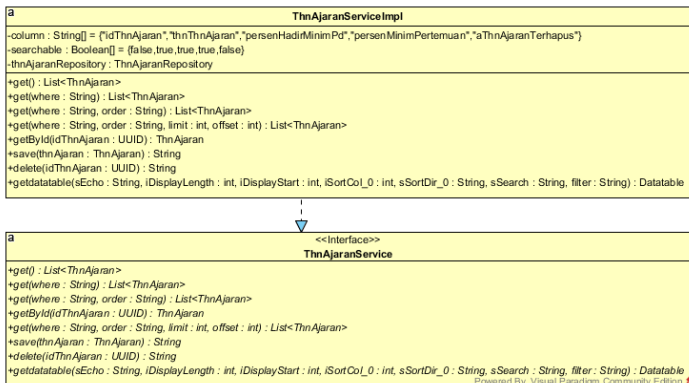
view. . Contoh salah satu diagram kelas pada paket controller dapat dilihat pada gambar 3.6.



Gambar 3.6. Diagram Kelas BeritaAcaraController

5. Paket Service

Paket Service merupakan kelas yang berfungsi sebagai pemroses data dan komputasi. Data yang diambil dari kelas *repository* diproses sesuai kasus penggunaan. Kelas ini sebagai penghubung antara kelas *controller* dan kelas *repository*. Contoh salah satu diagram kelas pada paket service dapat dilihat pada Gambar 3.7.

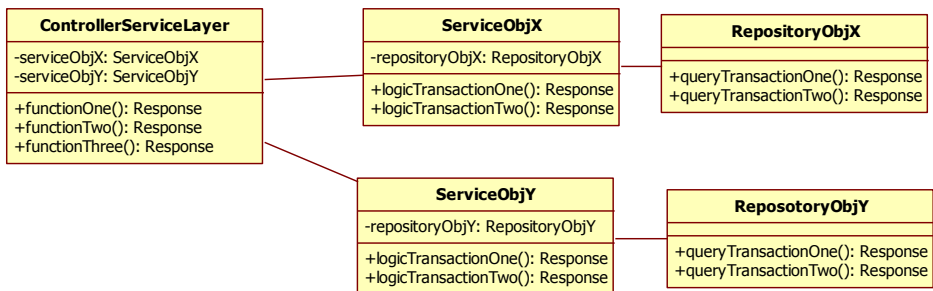


Gambar 3.7. Diagram Kelas ThnAjaranService

3.1.2.2. Penerapan Pola Perancangan *Service Layer*

Sistem informasi akademik versi replikasi ini telah menerapkan pola perancangan *Service Layer* pada *Domain Logic*-nya. *Service Layer* sendiri adalah pola perancangan yang menghubungkan *data source* dengan proses bisnis ataupun dengan komponen lain seperti tampilan antarmuka sistem.

Service Layer pada sistem ini dibuat dalam paket service yang berisi kelas-kelas proses-proses bisnis yang dibutuhkan oleh controller (domain). Pada paket service tidak memiliki hubungan dengan basis data. Untuk terhubung dengan basis data sistem ini memiliki paket Repository. Paket repository berisi kelas-kelas yang melakukan transaksi dengan basis data. Penerapan service layer dapat dilihat pada gambar 3.8.



Gambar 3.8. Diagram Kelas Pola Perancangan Service

3.1.3. Analisis Matrix Kualitas

Analisis karakteristik pada sistem informasi akademik memerlukan suatu matriks kualitas yang dibentuk dari kerangka model kualitas ISO 25023 dan ISO 25010. Subbab ini membahas analisis pembuatan matriks kualitas. Pada Tugas Akhir ini, subkarakteristik yang digunakan akan dibatasi pada subkarakteristik Maintainability. Sub-karakteristik Maintainability dipilih karena pada tugas akhir ini pengukuran kualitas difokuskan pada faktor internal khususnya *source code*.

Matriks parameter uji kualitas pemeliharaan terdiri atas lima sub-karakteristik dan 13 point penilaian. Sub-karakteristik pertama

adalah Modularity yang memiliki poin uji coupling of component conformance (MMo-1-G) dan Cyclomatic Complexity(MMo-2-S). Sub-karakteristik kedua adalah Reusability yang poin ujinya Reusable of Assets (MRe-1-G) dan conformance to coding rule (MRe-2-S). Sub-karakteristik ketiga adalah Analyzability dengan poin uji system log completeness conformance (MAn-1-G), diagnosis function effectiveness (Man-1-G) dan diagnosis function sufficiency conformance (Man-3-S). Sub-karakteristik keempat adalah testability yang memiliki parameter uji Test function completeness conformance (MTe-1-G), Autonomus testability (MTe-2-S) dan Test Restability (MTe-3-S). Sedangkan sub-karakteristik terakhir adalah Modifiability yang memiliki parameter uji modification efficiency (MMd-1-G), modification correctness (MMd-2-G) dan Modification capability (MMd-3-S).

Pada tugas akhir ini akan menggunakan empat dari lima sub-karakteristik yaitu Modularity, Reusability, Testability dan Analyzability. Terdapat satu sub-kualitas yaitu Modifiability yang tidak diukur pada tugas akhir ini. Modifiability tidak diukur karena membutuhkan *effort* yang cukup besar. Prosedur pengukuran sub-karakteristik Modifiability memerlukan beberapa *developer* bahasa pemrograman serupa yang cukup mahir.

Deskripsi karakteristik kualitas disajikan dalam tabel deskripsi (Tabel 3.1-Tabel3.10). Pada tabel deskripsi ada beberapa komponen untuk menjelaskan tentang pengukuran kualitas. Berikut keterangan dari masing-masing komponen

1. Nama Poin adalah nama poin kualitas.
2. Kode poin adalah kode kualitas yang diberikan oleh ISO 25023.
3. Sub-karakteristik adalah sub karakteristik dari poin kualitas.
4. Karakteristik Kualitas adalah kelompok karakteristik dari kualitas.
5. Bagian yang diuji adalah bagian dari sistem yang dilakukan pengujian.

6. Jumlah yang diuji adalah jumlah item dari sistem yang diuji. Jumlah dan kelompok yang diuji akan ditentukan sebelum uji kualitas.
7. Tujuan matriks adalah tujuan dari pengukuran sistem dengan poin kualitas ini.
8. Formula adalah rumus yang digunakan untuk mendapatkan nilai kualitas.
9. Keterangan formula adalah penjelasan variabel-variabel dari rumus.
10. Langkah pengukuran adalah langkah yang dilakukan untuk mendapatkan nilai penting dan nilai variabel.
11. Nilai Penting adalah nilai-nilai yang didapatkan untuk membantu mencari nilai variabel.
12. Kriteria adalah keterangan tambahan jika diperlukan tentang nilai dari variabel.

Tabel 3.1.Detail Pengujian Component Coupling

Komponen	Deskripsi
nama poin	coupling of component conformance
Kode poin	MMo-1-G
Sub-karakteristik Kualitas	Modularity
Karakteristik Kualitas	Maintainability
Bagian yang diuji	Source Code
Jumlah yang diuji	(Kelompok Interface)
tujuan matriks	mengukur tingkat independensi suatu komponen pada program
Formula	$X = A/B$
keterangan formula	X = hasil pengukuran
	A = Jumlah komponen yang diimplementasikan dengan dampak minimal pada komponen lainnya
	B = jumlah komponen yang harus <i>independent</i>

Komponen	Deskripsi
langkah pengukuran	<ol style="list-style-type: none"> 1. Jalankan plugin reverse engineering eclipse pada setiap kelas sehingga didapatkan diagram kelas dari masing-masing kelas. 2. Tentukan kelas yang akan kita hitung (misal kelas C). 3. Dari diagram kelas hitung jumlah kelas lain yang dependent terhadap kelas C. 4. Ulangi langkah 2 dan 3 hingga semua kelas terhitung. 5. Cari nilai A dengan menghitung jumlah kelas yang tidak memiliki kelas yang dependent terhadap kelas tersebut. 6. Dari kelas diagram dan kode program cari nilai B dengan menghitung jumlah kelas yang memiliki transaksi dengan basis data.
Nilai Penting	<ul style="list-style-type: none"> - Jumlah kelas Independent (HigherBetter) - jumlah kelas yang dependent terhadap suatu kelas. (SmallerBetter)
Kriteria	<ul style="list-style-type: none"> - Komponen yang memiliki dampak minimum adalah kelas yang tidak memerlukan perubahan dari kelas lain ketika kelas tersebut diubah. - Komponen yang seharusnya independent adalah kelas yang berisi proses transaksional dengan <i>database</i> atau kelas yang tidak memiliki proses bisnis.

Tabel 3.2.Detail Pengujian Cyclomatic complexity

Komponen	Deskripsi
nama poin	cyclomatic complexity
Kode poin	MMo-2-S
Sub-karakteristik Kualitas	Modularity
Karakteristik Kualitas	Maintainability

Komponen	Deskripsi
Bagian yang diuji	Source Code
Jumlah yang diuji	(Kelompok Implementasi)
tujuan matriks	Mengukur tingkat cyclomatic dari program
Formula	$X = 1 - A/B$
keterangan formula	x = hasil pengukuran
	a = jumlah fungsi yang memiliki kompleksitas cyclomatic melebihi trashhold
	b = jumlah fungsi.
langkah pengukuran	<ol style="list-style-type: none"> 1. Jalankan tools JHawk. 2. Masukkan file-file kelas yang akan dihitung nilai cyclomaticnya 3. Lihat hasil nilai cyclomatic fungsi dan jumlah fungsi dari masing-masing kelas. 4. Nilai B adalah jumlah fungsi dari seluruh kelas. 5. Nilai A adalah jumlah fungsi yang memiliki nilai cyclomatic diatas 10.
Nilai Penting	<ul style="list-style-type: none"> - Nilai Cyclomatic tiap-tiap fungsi (SmallerBetter) - fungsi dengan Nilai Cyclomatic tidak melebihi treshold (HigherBetter)
Kriteria	- Nilai Cyclomatic Threshold adalah 10

Tabel 3.3.Detail Pengujian Reusability of Assets

Komponen	Deskripsi
nama poin	Reusability of Assets
Kode poin	MRe-I-G
Sub-karakteristik Kualitas	Reusability
Karakteristik Kualitas	Maintainability
Bagian yang diuji	Source Code
Jumlah yang diuji	(Kelompok Interface)
tujuan matriks	mengukur tingkat penggunaan ulang komponen
Formula	$X = A/B$

Komponen	Deskripsi
keterangan formula	x = hasil pengukuran
	a = jumlah komponen dari sistem yang dibuat untuk dapat digunakan ulang
	b = jumlah komponen dari sistem
langkah pengukuran	<ol style="list-style-type: none"> 1. Jalankan plugin reverse engineering eclipse pada setiap kelas sehingga didapatkan diagram kelas dari masing-masing kelas. 2. Tentukan kelas yang akan kita hitung (misal kelas C). 3. Dari diagram kelas hitung jumlah kelas lain yang dependent terhadap kelas C. 4. Ulangi langkah 2 dan 3 hingga semua kelas terhitung. 5. Nilai A adalah jumlah kelas yang digunakan oleh lebih dari 1 kelas lain. 6. Nilai B adalah jumlah semua kelas.
nilai penting	<ul style="list-style-type: none"> - jumlah kelas yang di desain untuk dapat digunakan ulang (HigherBetter) - jumlah pemanggilan terhadap kelas (HigherBetter)
Kriteria	<ul style="list-style-type: none"> - Komponen yang dibuat untuk dapat digunakan ulang adalah kelas yang berisi proses lebih dari satu fitur atau tidak dibuat spesifik untuk membantu satu kelas tertentu.

Tabel 3.4. Detail Pengujian Conformance to coding rule

Komponen	Deskripsi
nama poin	conformance to coding rule
Kode poin	MRe-2-S
Sub-karakteristik Kualitas	Reusability
Karakteristik Kualitas	Maintainability
Bagian yang diuji	Source Code
Jumlah yang diuji	(Kelompok Interface dan Implementasi)

Komponen	Deskripsi
tujuan matriks	mengukur tingkat konsistensi penggunaan dalam menerapkan aturan penulisan program.
Formula	$X=A/B$
keterangan formula	X = hasil pengukuran
	A = jumlah komponen yang menerapkan aturan penulisan program
	B = jumlah komponen
langkah pengukuran	<ol style="list-style-type: none"> 1. Buka pengaturan plugin cekstyle eclipse. 2. Buat aturan pemrograman sesuai dengan aturan pemrograman yang telah ditentukan. 3. Cek aturan pemrograman dengan menjalankan plugin cekstyle pada semua kelas. 4. Nilai A adalah jumlah kelas yang tidak memiliki kesalahan sama sekali setelah dilakukan cek style. 5. Nilai B adalah jumlah semua kelas.
nilai penting	jumlah kelas yang menerapkan <i>code rule</i>
kriteria	<p>Aturan penulisan kode program adalah</p> <ul style="list-style-type: none"> - Penulisan nama paket menggunakan <i>upper camel case</i>. - Penulisan nama kelas menggunakan upper camel case. - Penulisan nama fungsi menggunakan lower camel case. - Penulisan nama variable menggunakan lower camel case. - Selalu tambahkan indentasi 1 tab setiap ada bracket baru - Deklarasi variable atau attribute kelas diletakkan bagian atas kelas sebelum deklarasi method - Hanya perbolehkan Deklarasi satu variable tiap baris.

Tabel 3.5.Detail Pengujian System log complateness conformace

Komponen	Deskripsi
nama poin	system log complateness conformace
Kode poin	MAN-1-G
Sub-karakteristik Kualitas	Analysability
Karakteristik Kualitas	Maintainability
Bagian yang diuji	Source Code
Jumlah yang diuji	(kelompok fitur)
tujuan matriks	mengukur seberapa jelas dan lengkap pencatatan proses program.
Formula	$X = A/B$
keterangan formula	X = hasil pengukuran
	A = jumlah log yang terekam
	B = jumlah log yang harusnya ada untuk audit sistem
langkah pengukuran	<ol style="list-style-type: none"> 1. Untuk mendapatkan log dapat menjalankan server log. 2. Nilai A adalah jumlah fitur yang menampilkan log ketika fitur diakses. 3. Nilai B adalah jumlah fitur.
nilai penting	jumlah log
Kriteria	- Log program seharusnya ada pada setiap akses ke fitur-fitur sistem.

Tabel 3.6.Detail Pengujian Efektifitas dan Diagnosa Fungsi

Komponen	Deskripsi
nama poin	diagnosis function effectiveness
Kode poin	Man-2-S
Sub-karakteristik Kualitas	Analysability
Karakteristik Kualitas	Maintainability

Bagian yang diuji	Source Code dan Dokumentasi
Jumlah yang diuji	(kelompok fitur)
tujuan matriks	Mengukur sejauh mana efektifitas implementasi fitur yang sesuai dengan kebutuhan dibandingkan dengan fitur yang telah diimplementasikan
Formula	$X = A/B$
keterangan formula	X = hasil pengukuran A = jumlah fitur yang menjawab kebutuhan B = jumlah fitur yang ada pada program
langkah pengukuran	<ol style="list-style-type: none"> 1. Dapatkan jumlah dan daftar fitur yang ada pada program. 2. Dapatkan jumlah dan daftar fitur yang didefinisikan pada spesifikasi kebutuhan program. 3. Nilai A adalah jumlah fitur yang ada pada program dan ada pada definisi kebutuhan program. 4. Nilai B adalah jumlah fitur yang ada pada program.
nilai penting	- Jumlah fitur dan kebutuhan
Kriteria	- Kebutuhan pada modul pembelajaran ini adalah kebutuhan yang telah didefinisikan pada penelitian sebelumnya.

Tabel 3.7.Detail Pengujian Diagnosa Kecukupan Fungsi

Komponen	Deskripsi
nama poin	diagnosis function sufficiency conformance
Kode poin	Man-3-S
Sub-karakteristik Kualitas	Analysability
Karakteristik Kualitas	Maintainability
Bagian yang diuji	Source Code + Dokumentasi
Jumlah yang diuji	(kelompok fitur)
tujuan matriks	mengukur sejauh mana fitur-fitur memenuhi spesifikasi kebutuhan.

Komponen	Deskripsi
Formula	$X = A/B$
keterangan formula	X = hasil pengukuran
	A = jumlah fitur yang telah dibuat berdasarkan kebutuhan
	B = jumlah fitur yang seharusnya dibuat sesuai dengan kebutuhan.
langkah pengukuran	<ol style="list-style-type: none"> 1. Dapatkan jumlah dan daftar fitur yang ada pada program. 2. Dapatkan jumlah dan daftar fitur yang didefinisikan pada spesifikasi kebutuhan program. 3. Nilai A adalah jumlah fitur yang didefinisikan pada dokumen kebutuhan dan diimplementasikan pada program. 4. Nilai B adalah jumlah fitur yang didefinisikan pada spesifikasi kebutuhan.
nilai penting	- Jumlah fitur dan kebutuhan
kriteria	- Kebutuhan pada modul pembelajaran ini adalah kebutuhan yang telah didefinisikan pada penelitian sebelumnya.

Tabel 3.8.Detail Pengujian Test function complateness conformance

Komponen	Deskripsi
nama poin	Test function complateness conformance
Kode poin	MTG-1-G
Sub-karakteristik Kualitas	Testability
Karakteristik Kualitas	Maintainability
Bagian yang diuji	Dokumentasi
Jumlah yang diuji	(Kelompok Uji Coba)
tujuan matriks	mengukur kelengkapan uji coba terhadap sistem
Formula	$X = A/B$

Komponen	Deskripsi
keterangan formula	x = hasil pengukuran
	a = jumlah uji coba yang dilakukan
	b = jumlah uji coba yang harusnya dilakukan
karakteristik keberhasilan	x disebut berhasil jika nilainya mendekati 1
langkah pengukuran	<ol style="list-style-type: none"> 1. Dapatkan daftar uji coba yang harusnya dilakukan dari dokumentasi program. 2. Nilai A adalah jumlah uji coba yang dilakukan 3. Nilai B adalah jumlah uji coba yang telah didefinisikan pada dokumentasi.
nilai penting	jumlah uji coba yang dilakukan
Kriteria	- Jumlah uji coba yang harusnya dilakukan adalah uji coba yang telah didefinisikan pada dokumentasi program.

Tabel 3.9.Detail Pengujian Autonomus testability

Komponen	Deskripsi
nama poin	Autonomus testability
Kode poin	MTG-2-S
Sub-karakteristik Kualitas	Testability
Karakteristik Kualitas	Maintainability
Bagian yang diuji	Source Code
Jumlah yang diuji	(Kelompok Uji Coba)
tujuan matriks	mengukur kemampuan Autonomus testability
Formula	$X = A/B$
keterangan formula	X = hasil pengukuran
	A = jumlah stub yang dapat berjalan pada dependency test
	B = jumlah uji coba yang harusnya dilakukan
karakteristik keberhasilan	x disebut berhasil jika nilainya mendekati 1

Komponen	Deskripsi
langkah pengukuran	Uji coba dilakukan dengan menghitung jumlah stub yang berjalan.
nilai penting	jumlah stub
kriteria	-

Tabel 3.10.Detail Pengujian Kemampuan Ulang Uji Coba

Komponen	Deskripsi
nama poin	Test Restartbility
Kode poin	MTG-3-S
Sub-karakteristik Kualitas	Testability
Karakteristik Kualitas	Maintainability
Bagian yang diuji	Dokumentasi
Jumlah yang diuji	(Kelompok Uji Coba)
tujuan matriks	mengukur kemudahan uji coba ulang setelah melakukan perbaikan sistem
Formula	$X = A/B$
keterangan formula	$X = \text{hasil pengukuran}$
	$A = \text{jumlah uji coba yang memiliki titik pause dan restart}$
	$B = \text{jumlah uji coba yang memiliki titik pause}$
langkah pengukuran	<ol style="list-style-type: none"> 1. Lakukan uji coba pada program sesuai dengan daftar uji coba. 2. Pada saat menjalankan langkah-langkah uji coba, hentikan uji coba/program. 3. Coba jalankan ulang uji coba. 4. Nilai A adalah jumlah uji coba yang tidak perlu melakukan uji coba dari awal ketika melakukan pause. 5. Nilai B adalah jumlah uji coba.
nilai penting	jumlah titik pause dan restart

Komponen	Deskripsi
Kriteria	<ul style="list-style-type: none"> - Uji coba dianggap dapat melakukan pause jika saat pengguna ingin melakukan perbaikan saat uji coba program dapat di hentikan sesaat. - Uji coba dianggap dapat melakukan restart jika pengguna dapat memulai kembali uji coba setelah melakukan perbaikan pada beberapa langkah sebelum perbaikan yang dilakukan dan tidak mengharuskan melakukan uji coba dari awal.

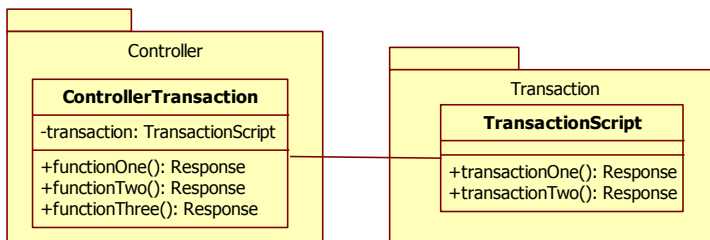
3.2. Perancangan

Subbab ini akan membahas tentang rancangan penerapan pola perancangan pada sistem informasi akademik ini. Pada sistem informasi akademik Modul Pembelajaran akan menerapkan pola perancangan pada kelompok Domain Logic. Pemilihan pola perancangan Domain Logic dikarenakan sistem informasi akademik versi replika ini telah menerapkan pola perancangan Service Layer yang merupakan bagian dari pola perancangan Domain Logic. Dengan dipilihnya pola perancangan Domain Logic diharapkan pada akhir tugas akhir ini dapat membandingkan empat pola perancangan Domain Logic. Pada dasarnya pola perancangan pada kelompok Domain Logic memiliki empat pola perancangan yaitu Transaction Script, Domain Model, Table Module dan Service Layer. sistem informasi akademik ini telah menerapkan pola perancangan Service Layer pada Domain Logic-nya. Tugas akhir ini akan menerapkan tiga pola perancangan lain yang belum diterapkan yaitu Transaction Script, Domain Model dan Table Module.

Tugas Akhir ini hanya berfokus pada penerapan pola perancangan pada lapisan Domain Logic sehingga perubahan hanya akan terjadi pada *source code* dan tidak akan berpengaruh dengan tampilan dan basis data sistem. Perancangan pada subbab akan direpresentasikan dengan diagram kelas.

3.2.1. Perancangan Transaction Script

Transaction Script adalah salah satu pola perancangan di kelompok Domain Logic. Pola perancangan ini adalah salah satu pola perancangan yang sederhana. Pola perancangan ini dianggap sederhana karena proses pada Transaction Script handle masing-masing proses. Satu proses yang dibutuhkan pada layer presentation dihandle oleh satu proses di Transaction Script.



Gambar 3.9. Rancangan Diagram Kelas Pola Perancangan Transaction Script

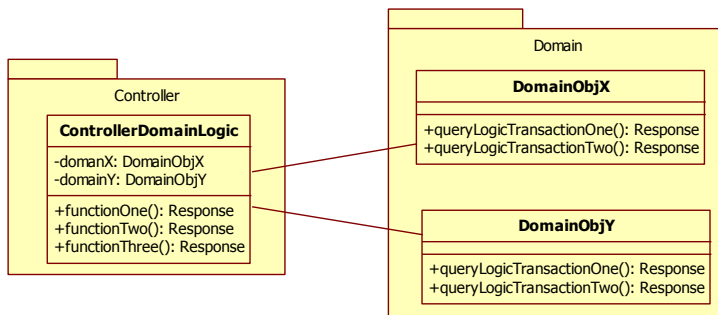
Dikarenakan tiap Transaction Script handle suatu proses langsung, maka untuk menerapkan pola perancangan Transaction Script ini memerlukan satu paket yang berisi kelas-kelas yang handle proses untuk masing-masing controller. Karena sistem ini memiliki 16 controller maka akan dibuat 16 kelas yang berisi proses untuk masing-masing controller sehingga controller dan paket Transaction Script ini memiliki hubungan 1 on 1 seperti yang ditunjukkan oleh gambar 3.9. Berikut kelas-kelas dalam paket transaction yang akan diimplementasikan untuk menerapkan pola perancangan Transaction Script

1. AbsensiTransaction
2. AturanPenggantiTransaction
3. BatasAmbilSksTransaction
4. BeritaAcaraTransaction
5. KrsTransaction
6. ManajemenKrsTransaction
7. PdTransaction

8. PembayaranTransaction
9. PembTransaction
10. PerwalianTransaction
11. PtkTransaction
12. RombelTransaction
13. SmtTransaction
14. StsKehadiranTransaction
15. TglSmtTransaction
16. ThnAjaranTransaction

3.2.2. Perancangan *Domain Model*

Domain Model adalah pola perancangan pada kelompok Domain Logic yang mengelompokkan proses-proses bisnis ke objek-objek individu yang merepresentasikan domain-domain masalah dari sistem. Setiap objek pada Domain Model akan berisi gabungan proses dari sebuah kelompok masalah.



Gambar 3.10. Rancangan Diagram Kelas Pola Perancangan Domain Model

Pada pengembangannya, controller yang membutuhkan proses logic akan memanggil domain yang sesuai. Tidak seperti Transaction Script, pada Domain Model ini tiap controller bisa memanggil lebih dari satu domain seperti yang ditampilkan pada gambar 3.10. Pada implementasinya akan dibuat paket domain yang berisi kelas-kelas untuk masing-masing domain yang

dibutuhkan. Kelas-kelas domain ini yang nantinya menyimpan proses logic yang dibutuhkan oleh controller.

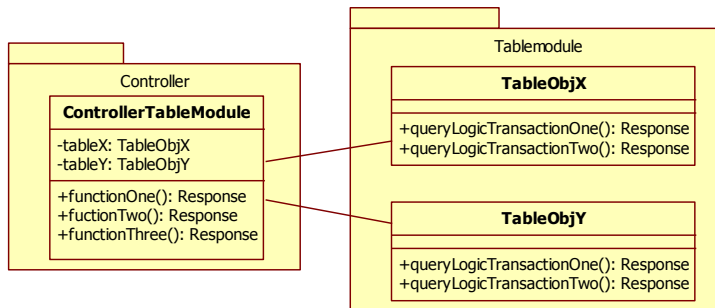
Domain sendiri akan dikelompokan sesuai dengan kelompok masalah yang telah ditentukan pada penelitian sebelumnya[3]. Domain-domain yang akan dibuat adalah

1. AnggotaRombel
2. AturanPengganti
3. BatasAmbilSks
4. Ipk
5. Ips
6. Krs
7. Kurikulum
8. ManajemenKrs
9. MenuPeran
10. MK
11. Pd
12. Pembayaran
13. Pemb
14. PembSatMan
15. PendidikPengajar
16. Pengguna
17. Peran
18. PeranPengguna
19. PertemuanPembelajaran
20. PrasyaratMK
21. PresensiPd
22. PresensiPengajar
23. Ptk
24. Rombel
25. SatMan
26. SatManMK
27. Smt
28. StatusKuisisioner
29. StsKehadiran
30. TglSmt

31. ThnAjaran

3.2.3. Perancangan *Table Module*

Table Module adalah pola perancangan pada kelompok Domain Logic yang mengelompokkan proses-proses bisnis berdasarkan table yang telah dibuat. Setiap Table Module akan berisi proses-proses yang dibutuhkan controller untuk mengambil data dari table basis data.



Gambar 3.11. Rancangan Diagram Kelas Pola Perancangan Table Module

Pada implementasinya, akan dibuat paket module yang berisi kelas-kelas Table Module seperti gambar 3.11. Karena Table Module dibuat berdasarkan table yang telah dibuat maka kelas-kelas Table Module yang dibutuhkan adalah

1. AnggotaRombelModul
2. AturanPenggantiModul
3. BatasAmbilSksModul
4. IpkModul
5. IpsModul
6. KrsModul
7. KurikulumModul
8. MenuPeranModul
9. MKModul
10. PdModul
11. PembayaranModul

12. PembModul
13. PembSatManModul
14. PendidikPengajarModul
15. PenggunaModul
16. PeranModul
17. PeranPenggunaModul
18. PertemuanPembelajaranModul
19. PrasyaratMKModul
20. PresensiPdModul
21. PresensiPengajarModul
22. PtkModul
23. RombelModul
24. SatManMKModul
25. SatManModul
26. SmtModul
27. StatusKuisisionerModul
28. StsKehadiranModul
29. TglSmtModul
30. ThnAjaranModul

Ada salah satu masalah dalam penerapan pola perancangan ini yaitu beberapa proses yang dibutuhkan controller tidak hanya berasal dari satu table (perlu ada relasi dengan table lain). Untuk mengatasi masalah tersebut maka proses yang dibutuhkan akan ditempatkan pada salah satu Table Module yang memiliki peran lebih besar seperti menjadi data utama yang dibutuhkan.

3.2.4. Perbandingan Pengelompokan Domain Logic

Salah satu hal penting dari penerapan pola perancangan domain logic adalah pengelompokan domain logic terhadap proses-proses yang dibutuhkan oleh controller. Pada Bab 3.1.2 , 3.2.1, 3.2.2 dan 3.2.3 telah dibahas bagaimana setiap pola perancangan menerapkan pengelompokan domain logicnya. Untuk melihat perbandingan pengelompokan pola perancangan lihat table 3.11.

Jika dilihat dari perbandingan pengelompokan (table 3.11), service layer memiliki total 31 kelompok domain logic

sama seperti domain model. Table Module memiliki 30 pengelompokan domain logic. Pengelompokan terkecil ada pada pola perancangan Transaction Script dengan hanya 16 kelompok sesuai dengan jumlah controller.

Tabel 3.11. Perbandingan Kelompok Domain Logic

No	Service Layer	Transaction Script	Domain Model	Table Module
1	AnggotaRombelService	AbsensiTransaction	AnggotaRombel	MenuPeranModul
2	AturanPenggantiService	AturanPengantiTransaction	AturanPengga nti	PeranPenggunaModul
3	BatasAmbilSksService	BatasAmbilSksTransaction	BatasAmbilSks	KrsModul
4	IpkService	BeritaAcaraTransaction	Ipk	ThnAjaranModul
5	IpsService	KrsTransaction	Ips	SmtModul
6	KrsService	ManajemenKrsTransaction	Krs	SatManModul
7	KurikulumService	PdTransaction	Kurikulum	PendidikPengajarModul
8	ManajemenKrsService	PembayaranTransaction	ManajemenKrs	SatManMKModul
9	MenuPeranService	PembTransaction	MenuPeran	TglSmtModul
10	MKService	PerwalianTransaction	MK	PrasyaratMKModul
11	PdService	PtkTransaction	Pd	PenggunaModul
12	PembayaranService	RombelTransaction	Pembayaran	PdModul
13	PembService	SmtTransaction	Pemb	PertemuanPembelajaranModul
14	PembSatManService	StsKehadiranTransaction	PembSatMan	RombelModul
15	PendidikPengajarService	TglSmtTransaction	PendidikPengajar	KurikulumModul
16	PenggunaService	ThnAjaranTransaction	Pengguna	PresensiPengajarModul

No	Service Layer	Transaction Script	Domain Model	Table Module
17	PeranService		Peran	BatasAmbilSksModul
18	PeranPenggunaService		PeranPengguna	PembModul
19	PertemuanPembelajaranService		PertemuanPembelajaran	PeranModul
20	PrasyaratMKService		PrasyaratMK	StsKehadiranModul
21	PresensiPdService		PresensiPd	PtkModul
22	PresensiPengajarService		PresensiPengajar	PembSatManModul
23	PtkService		Ptk	PresensiPdModul
24	RombelService		Rombel	IpkModul
25	SatManService		SatMan	StatusKuisisionerModul
26	SatManMKService		SatManMK	AnggotaRombelModul
27	SmtService		Smt	IpsModul
28	StatusKuisisionerService		StatusKuisisioner	PembayaranModul
29	StsKehadiranService		StsKehadiran	AturanPenggantiModul
30	TglSmtService		TglSmt	MKModul
31	ThnAjaranService		ThnAjaran	

BAB IV IMPLEMENTASI SISTEM

Bab ini membahas mengenai implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Implementasi yang dijelaskan adalah bagaimana menerapkan pola perancangan *Transaction Script*, *Domain Model* dan *Table Module* pada modul pembelajaran. Bahasa pemrograman yang digunakan untuk implementasi adalah bahasa pemrograman Java menggunakan kerangka kerja Spring.

4.1. Lingkungan Pengembangan Sistem

Lingkungan pengembangan sistem yang digunakan untuk mengembangkan Tugas Akhir ini dilakukan pada lingkungan dan kakas sebagai berikut.

1. Sistem operasi Windows 8 Professional 64 bit.
2. StarUML digunakan untuk membuat diagram UML
3. pgAdmin III digunakan untuk mengelola basis data.
4. Eclipse Mars untuk IDE.
5. Putty untuk remote Server.

4.2. Penerapan Paket Controller, Validator dan Data

Penerapan pola perancangan pada sistem ini tidak serta merta mengganti semua komponen dari sistem. Ada beberapa komponen sistem yang tidak diubah. Komponen yang tidak diubah adalah Paket Controller dan Paket Validator. Semua kelas-kelas pada paket controller dan validator tetap sama dengan sistem lama hanya berbeda pada pemanggilan kelas *Domain Logic*nya bergantung dengan pola perancangan yang diterapkan.

Selain ada komponen yang tetap, juga ada komponen yang mengalami sedikit perubahan. Komponen yang sebelumnya ada dan mengalami sedikit modifikasi adalah kelas-kelas pada paket data. Kelas-kelas pada paket data sebelumnya telah ada pada paket service. Kelas pada paket data sebelumnya ada pada paket service namun tidak berfungsi sebagai *Service Layer*. Kelas-kelas ini

hanya kelas untuk merepresentasikan data dan mengirimnya ke view sistem.

4.3. Penerapan Pola Perancangan *Transaction Script*

Pada pola perancangan *Transaction Script*, dibuat kelas-kelas yang berelasi 1 on 1 dengan kelas controller. *Transaction Script* berisi fungsi-fungsi yang menjalankan proses yang dibutuhkan controller. Penerapan *Transaction Script* sendiri membuat kelas-kelas yang dibungkus pada paket Transaction. Setiap *Transaction Script* memiliki 2 kelas yaitu kelas interface dan kelas implementasi. Berikut kelas-kelas yang terdapat pada paket transaction

1. **AbsensiTransaction dan AbsensiTransactionImpl**

Merupakan kelas *Transaction Script* yang digunakan untuk handle proses untuk AbsensiController

2. **AturanPenggantiTransaction dan aturanPenggantiTransactionImpl**

Merupakan kelas *Transaction Script* yang digunakan untuk handle proses untuk AturanPenggantiController

3. **BatasAmbilSksTransaction dan BatasAmbilSksTransactionImpl**

Merupakan kelas *Transaction Script* yang digunakan untuk handle proses untuk BatasAmbilSksController

4. **BeritaAcaraTransaction dan BeritaAcaraTransactionImpl**

Merupakan kelas *Transaction Script* yang digunakan untuk handle proses untuk BeritaAcaraController

5. **KrsTransaction dan KrsTransactionImpl**

Merupakan kelas *Transaction Script* yang digunakan untuk handle proses untuk KrsController

**6. ManajemenKrsTransaction dan
ManajemenKrsTransactionImpl**

Merupakan kelas *Transaction Script* yang digunakan untuk handle proses untuk ManajemenKrsController

7. PdTransaction dan PdTransactionImpl

Merupakan kelas *Transaction Script* yang digunakan untuk handle proses untuk PdController

**8. PembayaranTransaction dan
PembayaranTransactionImpl**

Merupakan kelas *Transaction Script* yang digunakan untuk handle proses untuk PembayaranController

9. PembTransaction dan PembTransactionImpl

Merupakan kelas *Transaction Script* yang digunakan untuk handle proses untuk PembController

10. PerwalianTransaction dan PerwalianTransactionImpl

Merupakan kelas *Transaction Script* yang digunakan untuk handle proses untuk AbsensiController

11. PtkTransaction dan PtkTransactionImpl

Merupakan kelas *Transaction Script* yang digunakan untuk handle proses untuk PtkController

12. RombelTransaction dan RombelTransactionImpl

Merupakan kelas *Transaction Script* yang digunakan untuk handle proses untuk RombelController

13. SmtTransaction dan SmtTransactionImpl

Merupakan kelas *Transaction Script* yang digunakan untuk handle proses untuk SmtController

**14. StsKehadiranTransaction dan
StsKehadiranTransactionImpl**

Merupakan kelas *Transaction Script* yang digunakan untuk handle proses untuk StsKehadiranController

15. TglSmtTransaction dan TglSmtTransactionImpl

Merupakan kelas *Transaction Script* yang digunakan untuk handle proses untuk TglSmtController

16. ThnAjaranTransaction dan ThnAjaranTransactionImpl

Merupakan kelas *Transaction Script* yang digunakan untuk handle proses untuk ThnAjaranController

Berikut salah satu contoh proses pemanggilan *Transaction Script* pada controller.

	...
40	@Autowired
41	private PdTransaction transaction;
	...
74	@RequestMapping(value = "/simpan", method = RequestMethod. POST)
75	public @ResponseBody AjaxResponse simpan(@Valid @ModelAttribute("tglSmt") Pd pd,
76	BindingResult result, Map<String, Object> model,
77	@RequestParam("idPtk") UUID idPtk) {
78	AjaxResponse response = new AjaxResponse();
79	if (idPtk!= null)pd.setPtk(transaction.ptkGetById(idPtk));
80	if (result.hasErrors()) {
81	response.setStatus("error");
82	List<FieldError> fieldError = result.getFieldErrors();
83	String message = "";
84	if (fieldError.get(0).isBindingFailure()) message += "Salah satu input tiak valid";
85	else message += fieldError.get(0).getDefaultMessage();
86	for (int i=1;i<fieldError.size();i++){

87	if (fieldError.get(i).isBindingFailure()) message += " Salah satu input tiak valid";
88	else message += " " + fieldError.get(i).getDefaultMessage();
89	}
90	response.setMessage(message);
91	response.setData(fieldError);
92	return response;
93	}
94	response.setStatus("ok");
95	response.setData(transaction.pdSave(pd));
96	if (response.getData() != null) response.setMessage("Data berhasil disimpan");
97	else
98	{
99	response.setStatus("error");
100	response.setMessage("NIM terpakai");
101	}
102	return response;
103	}
	...

Kode Sumber 4.1.Fungsi Simpan pada Kelas PdTransaction

108	@Override
109	public String pdSave(Pd pd) {
110	String where = "";
111	if (pd.getIdPd() != null) where += " AND pd.idPd != "" + pd.getIdPd() + "";
112	List<Pd> queryResult = pdGet("pd.nimPd = "" + pd.getNimPd() + "" + where, "", -1, -1);
113	if (queryResult.size() > 0) return null ;
114	else if (pd.getIdPd() != null)
115	{
116	pdUpdate(pd);
117	return pd.getIdPd().toString();
118	}
119	else

120	{
121	return pdInsert(pd).toString();
122	}
123	}

Kode Sumber 4.2.Fungsi pdSave pada Kelas PdTransaction

Potongan program diatas adalah proses menyimpan peserta didik. Proses berawal dari PdController.java yang mendeklarasikan objek transaction yang merupakan PdTransaction (line40). Objek transaction yang telah dideklarasikan digunakan untuk memanggil fungsi-fungsi PdTransaction seperti pada kode sumber 4.8 line 79 dan 95. Sedangkan pada kode sumber 4.9 menunjukan proses penyimpanan pada kelas PdTransaction.

Penerapan pola perancangan transaction script membuat perubahan dependency controller serta pemetaan proses. Pemetaan dan dependency controller ditampilkan pada tabel 4.1.

Tabel 4.1. Pemetaan Kelas Controller dengan Kelas Transaction Scriptnya

No	Nama Controller	Nama Transaction Script
1	AbsensiController	AbsensiTransaction
2	AturanPenggantiController	AturanPenggantiTransaction
3	BatasAmbilSksController	BatasAmbilSksTransaction
4	BeritaAcaraController	BeritaAcaraTransaction
5	KrsController	KrsTransaction
6	ManajemenKrsController	ManajemenKrsTransaction
7	PdController	PdTransaction
8	PembayaranController	PembayaranTransaction
9	PembController	PembTransaction
10	PerwalianController	PerwalianTransaction
11	PtkController	PtkTransaction
12	RombelController	RombelTransaction

No	Nama Controller	Nama Transaction Script
13	SmtController	SmtTransaction
14	StsKehadiranController	StsKehadiranTransaction
15	TglSmtController	TglSmtTransaction
16	ThnAjaranController	ThnAjaranTransaction

Penerapan pola perancangan Transaction Script tentunya merubah kompleksitas program. Berikut kompleksitas program setelah menerapkan pola perancangan transaction script

- Jumlah Kelas: 57 Kelas
- Total Line of Code: 5878 Line
- Total Method: 454 Method
- Total Cyclomatic: 1331

4.4. Penerapan Pola Perancangan *Domain Model*

Pada pola perancangan *Domain Model*, dibuat kelas-kelas yang menyimpan proses bisnis dari suatu kelompok masalah atau domain masalah. Penerapan pola perancangan *Domain Model* ini sendiri dengan membuat kelas-kelas yang dibungkus pada paket domain. Setiap *Domain Model* terdiri dari satu interface dan satu kelas implementasinya.

Penerapan pola perancangan Domain Model membuat perubahan dependency controller serta pemetaan proses. Pemetaan dan dependency controller ditampilkan pada tabel 4.2.

Tabel 4.2. Pemetaan Kelas Controller dengan Kelas Domain Model

No	Nama Controller	Domain Model
1	AbsensiController	PembDm
		TglSmtDm
		ThnAjaranDm
		SmtDm
		PendidikPengajarDm

No	Nama Controller	Domain Model
		KrsDm
		SatManDm
		StsKehadiranDm
		PertemuanPembelajaranDm
		PresensiPengajarDm
		PdDm
		PresensiPdDm
2	AturanPenggantiController	TglSmtDm
		SatManDm
		AturanPenggantiDm
3	BatasAmbilSksController	BatasAmbilSksDm
4	BeritaAcaraController	PembDm
		TglSmtDm
		PendidikPengajarDm
		PertemuanPembelajaranDm
5	KrsController	KrsDm
		TglSmtDm
		PdDm
		PembDm
		AnggotaRombelDm
6	ManajemenKrsController	AturanPenggantiDm
		TglSmtDm
		ThnAjaranDm
		SmtDm
		PembDm
		PdDm
		ManajemenKrsDm

No	Nama Controller	Domain Model
		AturanPenggantiDm
		IpsDm
		IpkDm
		BatasAmbilSksDm
		PembayaranDm
		PrasyaratMKDm
		SatManMKDm
		PeranPenggunaDm
7	PdController	PdDm
		PtkDm
8	PembayaranController	PembayaranDm
		ThnAjaranDm
		SmtDm
		SatManDm
		TglSmtDm
		PdDm
9	PembController	PembDm
		MKDm
		TglSmtDm
		KurikulumDm
		PendidikPengajarDm
		PtkDm
		PdDm
		KrsDm
		SatManDm
		PembSatManDm
		RombelDm

No	Nama Controller	Domain Model
10	PerwalianController	PtkDm
		PdDm
11	PtkController	PtkDm
		PdDm
12	RombelController	RombelDm
		AnggotaRombelDm
		PdDm
		SatManDm
13	SmtController	SmtDm
14	StsKehadiranController	StsKehadiranDm
15	TglSmtController	TglSmtDm
		ThnAjaranDm
		SmtDm
16	ThnAjaranController	ThnAjaranDm

Domain dikelompokan berdasarkan kelompok masalah yang telah ditentukan pada penelitian sebelumnya. Penerapan pola perancangan Domain Model tentunya merubah kompleksitas program. Berikut kompleksitas program setelah menerapkan Domain Model

- Jumlah Kelas: 87 Kelas
- Total Line of Code: 6073 Line
- Total Method: 576 Method
- Total Cyclomatic: 1310

4.5. Penerapan Pola Perancangan *Table Module*

Pada pola perancangan *Table Module*, dibuat kelas-kelas yang berelasi 1 on 1 dengan tabel pada basis data. *Table Module* berisi fungsi-fungsi yang menjalankan proses interaksi dengan tiap tabel pada basis data yang dibutuhkan controller. Penerapan *Table*

Module sendiri membuat kelas-kelas yang dibungkus pada *tablemodule*. Setiap *Table Module* memiliki 2 kelas yaitu kelas interface dan kelas implementasi.

Kelas-kelas pada *Table Module* memiliki beberapa fungsi umum karena fungsi-fungsi tersebut adalah fungsi dasar dalam transaksi dengan tabel basis data. Fungsi-fungsi umum pada *table module* ditampilkan pada tabel 4.1. Contoh implementasi fungsi-fungsi umum pada *Table Module* terdapat pada kode sumber 4.41. namun pada *Table Module* fungsi-fungsi yang ada tidak hanya fungsi umum. Ada juga fungsi-fungsi lain yang menjalankan proses khusus dari transaksi dengan *table*.

**Tabel 4.3.Fungsi-fungsi Umum pada Kelas-Kelas Paket
Tablemodule**

Return value	Nama fungsi dan penjelasan
List<Obj>	get(String where, String order, int limit, int offset)
	Mengambil data pada tabel dengan filter where, urutan order batasan jumlah data limit dan mulai dari baris offset.
Obj	getById(UUID idTabel)
	Mengambil data dari tabel dengan ID yang sama dengan idTabel.
UUID	insert(Obj data)
	Menyimpan atau mengubah data pada tabel dan mengembalikan id data.
void	update(Obj data)
	Mengubah data pada tabel.
void	delete(Obj obj)
	Menghapus data pada tabel.
Long	count(String where)
	Menghitung data pada tabel dengan filter where
Datatable	Getdatatable (String sEcho, int iDisplayLength, int iDisplayStart, int iSortCol_0, String sSortDir_0, String sSearch, String filter)
	Mengambil data pada table dengan filter dan result berupa datatable
String	save(obj data)
	Menyimpan data pada tabel

1	@Repository
2	@Transactional
3	@Component
4	public class AnggotaRombelModulImpl implements AnggotaRombelModul {
5	@Autowired
6	private SessionFactory sessionFactory;
7	private String[] column = {"anggotaRombel.idAnggotaRombel", "pd.nimPd", "pd.nmPd", "pd.angkatanPd"};
8	private Boolean[] searchable = {false,true,true,true,false};
9	@Override
10	public List<AnggotaRombel> get(String where, String order, int limit, int offset) {
11	String dbWhere = "";
12	String dbOrder = "";
13	if (where != "") dbWhere = " WHERE "+where;
14	if (order != "") dbOrder = " ORDER BY "+order;
15	Query query = sessionFactory.getCurrentSession().createQuery("select anggotaRombel from AnggotaRombel " + "anggotaRombel join anggotaRombel.pd pd join anggotaRombel.rombel rombel" +dbWhere+dbOrder);
16	if (limit != -1 && limit>0) {
17	query.setFirstResult(offset);
18	if (offset < 0) offset = 0;
19	query.setMaxResults(limit);
20	}
21	return query.list();
22	}
23	@Override
24	public AnggotaRombel getById(UUID idAnggotaRombel) {
25	List<AnggotaRombel> queryResult = sessionFactory.getCurrentSession().createQuery("select anggotaRombel from AnggotaRombel " + "anggotaRombel join anggotaRombel.pd pd join anggotaRombel.rombel rombel " + "WHERE anggotaRombel.idAnggotaRombel = +idAnggotaRombel.toString()+""").list();

26	if (queryResult.size()==0) return null ;
27	return queryResult.get(0);
28	}
29	@Override
30	public UUID insert(AnggotaRombel ptk) {
31	Session session = sessionFactory.openSession();
32	Transaction tx = session.beginTransaction();
33	UUID insertId= (UUID)session.save(ptk);
34	tx.commit();
35	session.flush();
36	session.close();
37	return insertId;
38	}
39	@Override
40	public void update(AnggotaRombel anggotaRombel){
41	Session session = sessionFactory.openSession();
42	Transaction tx = session.beginTransaction();
43	session.update(anggotaRombel);
44	tx.commit();
45	session.flush();
46	session.close();
47	}
48	@Override
49	public void delete(AnggotaRombel anggotaRombel){
50	Session session = sessionFactory.openSession();
51	Transaction tx = session.beginTransaction();
52	session.delete(anggotaRombel);
53	tx.commit();
54	session.flush();
55	session.close();
56	}
57	@Override
58	public long count(String where) {
59	String dbWhere = "";
60	if (where != "") dbWhere = " WHERE "+where;
61	Query query = sessionFactory.getCurrentSession().createQuery("select count(*) from AnggotaRombel " + "anggotaRombel join

	<code>anggotaRombel.pd pd join anggotaRombel.rombel rombel "+</code> <code>dbWhere);</code>
62	<code>Long count = (Long)query.uniqueResult();</code>
63	<code>return count;</code>
64	<code>}</code>
65	<code>@Override</code>
66	<code>public String save(AnggotaRombel anggotaRombel){</code>
	<code>List<AnggotaRombel> listAnggotaRombel =</code> <code>get("pd.idPd='"+anggotaRombel.getPd().getIdPd()+"' AND</code> <code>rombel.idRombel =</code>
67	<code>"'+anggotaRombel.getRombel().getIdRombel()+''", "",-1,-1);</code>
68	<code>if(listAnggotaRombel.size(>0)</code>
69	<code>{</code>
70	<code>return null;</code>
71	<code>}</code>
72	<code>else if(anggotaRombel.getIdAnggotaRombel() != null)</code>
73	<code>{</code>
74	<code>update(anggotaRombel);</code>
75	<code>return anggotaRombel.getIdAnggotaRombel().toString();</code>
76	<code>}</code>
77	<code>else</code>
78	<code>{</code>
79	<code>return insert(anggotaRombel).toString();</code>
80	<code>}</code>
81	<code>}</code>
82	<code>@Override</code>
	<code>public Datatable getdatatable(String sEcho, int</code> <code>iDisplayLength,int iDisplayStart, int iSortCol_0, String</code>
83	<code>sSortDir_0,String sSearch, String filter) {</code>
	<code>DatatableExtractParams parameter = new</code> <code>DatatableExtractParams(sSearch, this.column, this.searchable,</code>
84	<code>iSortCol_0, sSortDir_0);</code>
85	<code>Datatable rombelDatatable= new Datatable();</code>
86	<code>rombelDatatable.setsEcho(sEcho);</code>
87	<code>String dbFilter = "";</code>
88	<code>if(filter != null && !filter.equals("")) dbFilter+=" AND "+filter;</code>

	List<AnggotaRombel> queryResult =
89	get("(" + parameter.getWhere() + ")"+dbFilter,
90	parameter.getOrder(), iDisplayLength, iDisplayStart);
91	List<String[]> aData = new ArrayList<String[]>();
92	for (AnggotaRombel anggotaRombel : queryResult){
93	String[] anggotaRombelString = new String[5];
	anggotaRombelString[0] =
	anggotaRombel.getIdAnggotaRombel().toString();
	anggotaRombelString[1] =
	String.valueOf(anggotaRombel.getPd().getNimPd())anggotaRo
94	mbelString[2] =
	String.valueOf(anggotaRombel.getPd().getNmPd());
	anggotaRombelString[3] =
95	String.valueOf(anggotaRombel.getPd().getAngkatanPd());
	anggotaRombelString[4] =
96	String.valueOf(anggotaRombel.getIdAnggotaRombel());
97	aData.add(anggotaRombelString);
98	}
99	rombelDatatable.setAaData(aData);
100	rombelDatatable.setiTotalRecords(count(filter));
	rombelDatatable.setiTotalDisplayRecords(count("
101	" + parameter.getWhere() + ") AND " + filter));
102	return rombelDatatable;
103	}
104	}

Kode Sumber 4.3.Implementasi Salah Satu Kelas Pada *Table Module*

Penerapan pola perancangan Table Module membuat perubahan dependency controller serta pemetaan proses. Pemetaan dan dependency controller ditampilkan pada tabel 4.4

Tabel 4.4. Pemetaan Kelas Controller dengan Kelas Table Module

No	Nama Controller	Table Module
1	AbsensiController	KrsModul
		PdModul
		PembModul

No	Nama Controller	Table Module
		PendidikPengajarModul
		PertemuanPembelajaranModul
		PresensiPdModul
		PresensiPengajarModul
		SatManModul
		SmtModul
		StsKehadiranModul
		TglSmtModul
		ThnAjaranModul
2	AturanPenggantiController	TglSmtModul
		SatManModul
		AturanPenggantiModul
3	BatasAmbilSksController	BatasAmbilSksModul
4	BeritaAcaraController	PembModul
		PendidikPengajarModul
		PertemuanPembelajaranModul
		TglSmtModul
5	KrsController	AnggotaRombelModul
		KrsModul
		PdModul
		PembModul
		TglSmtModul
6	ManajemenKrsController	AturanPenggantiModul
		BatasAmbilSksModul
		IpkModul
		IpsModul
		KrsModul

No	Nama Controller	Table Module
		PdModul
		PembModul
		PembayaranModul
		PeranPenggunaModul
		PrasyaratMKModul
		SatManMKModul
		SmtModul
		TglSmtModul
		ThnAjaranModul
7	PdController	PdModul
		PtkModul
8	PembayaranController	PdModul
		PembayaranModul
		SatManModul
		SmtModul
		TglSmtModul
		ThnAjaranModul
9	PembController	KrsModul
		KurikulumModul
		MKModul
		PdModul
		PembModul
		PembSatManModul
		PendidikPengajarModul
		PtkModul
		RombelModul
		SatManModul

No	Nama Controller	Table Module
		TglSmtModul
10	PerwalianController	PdModul
		PtkModul
11	PtkController	PtkModul
12	RombelController	AnggotaRombelModul
		PdModul
		RombelModul
		SatManModul
13	SmtController	SmtModul
14	StsKehadiranController	StsKehadiranModul
15	TglSmtController	SmtModul
		TglSmtModul
		ThnAjaranModul
16	ThnAjaranController	ThnAjaranModul

Penerapan pola perancangan Domain Model tentunya merubah kompleksitas program. Berikut

- Jumlah Kelas: 85 Kelas
- Total Line of Code: 5591 Line
- Total Method: 450 Method
- Total Cyclomatic: 1141

4.6. Proses Deployment Program

Proses deployment program adalah langkah yang harus dilakukan setelah penerapan pola perancangan selesai. Proses deployment adalah mengunggah program ke server. Server pada sistem informasi akademik ini menggunakan server virgo yang telah diberi Osgi sebagai *bundle management*. Karena sistem informasi akademik ini sebelumnya telah direfactor menjadi program berbasis osgi bundle, uji coba program setelah

implementasi harus dilakukan langsung pada server virgo. Hal tersebut cukup menyita waktu karena proses unggah program cukup rumit.

Proses setelah implementasi menuju unggah program memiliki dua macam proses. Proses pertama adalah upload artifac ke server virgo. Proses pertama digunakan untuk upload modul-modul Sia-domain, Sia-modul-domain, Sia-data, Sia-plugin, Sia-service dan sia-web. Langkah-langkah untuk proses pertama adalah sebagai berikut

1. Lakukan maven-build pada modul
2. Buka halaman admin virgo pada 10.151.44.20:9191/admin
3. Masukan *username* dan *password*
4. Masuk ke tab artifac
5. Upload file-file .jar dan .war pada folder target masing-masing modul

Proses kedua adalah proses yang digunakan untuk mengupload modul pembelajaran. Upload modul pembelajaran tidak langsung ke server virgo namun langsung pada osgi. Langkah-langkah proses kedua adalah sebagai berikut

1. Buka web sistem informasi akademik pada 10.151.44.20:9191/sia
2. Login dengan akun administrator@its.ac.id
3. Buka menu tambah modul
4. Upload file .jar modul pembelajaran yang telah di build
5. Tentukan hak akses masing-masing role
6. Pilih icon dan upload gambar modul
7. Pilih selesai.

4.7. Perbandingan Hasil Penerapan Pola Perancangan

Setelah selesai merancangan dan menerapkan pola perancangan didapatkan bahwa desain arsitektur penerapan empat pola perancangan relative sama. Perbedaan terbesar dari struktur

empat pola perancangan adalah penerapan service layer. Service layer menggunakan satu paket yang berisi service dan satu paket sebagai repository. Struktur program yang hampir samat tiap pola perancangannya bukanlah suatu masalah, karena memang tujuan dari domain logic adalah mengelompokkan logic sehingga strukturnya.

Perbedaan dari penerapan pola-pola perancangan pada domain logic adalah *dependency* dari controller. Controller yang berperan sebagai pemanggil proses pada domain logic akan melakukan pemanggilan yang berbeda-beda jika penerapan pada domain logic juga berbeda. Perbedaan dependency controller dapat terlihat pada tabel 4.5. Penerapan dari controller hingga pola-pola perancangan telah disesuaikan dengan rancangan program.

Tabel 4.5. Pemetaan Kelas Controller dengan Kelas-kelas pada Empat Pola Perancangan

Nama Controller	Service Layer	Domain Model	Table Module	Transaction Script
Absensi Controller	KrsService	PembDm	KrsModul	Absensi Transaction
	PdService	TglSmtDm	PdModul	
	PembSatMan Service	ThnAjaranDm	PembModul	
	PendidikPengajar Service	SmtDm	PendidikPengajarModul	
	PertemuanPembelajaranService	PendidikPengajarDm	PertemuanPembelajaranModul	
	PresensiService	KrsDm	PresensiPdModul	
	PtkService	SatManDm	PresensiPengajarModul	
	SatManService	StsKehadiranDm	SatManModul	
	SmtService	PertemuanPembelajaranDm	SmtModul	
	StsKehadiranService	PresensiPengajarDm	StsKehadiranModul	
	TglSmtService	PdDm	TglSmtModul	

Nama Controller	Service Layer	Domain Model	Table Module	Transaction Script
	ThnAjaranService	PresensiPdDm	ThnAjaranModul	
Aturan Pengganti Controller	AturanPenggantiService	TglSmtDm	TglSmtModul	Aturan Pengganti Transaction
	SmtService	SatManDm	SatManModul	
	TglSmtService	AturanPenggantiDm	AturanPenggantiModul	
	ThnAjaranService			
BatasAmbil SksController	BatasAmbilSksService	BatasAmbilSksDm	BatasAmbilSksModul	BatasAmbilSks Transaction
BeritaAcara Controller	PembService	PembDm	PembModul	BeritaAcara Transaction
	PendidikPengajarService	TglSmtDm	PendidikPengajarModul	
	PertemuanPembelajaranService	PendidikPengajarDm	PertemuanPembelajaranModul	
	PtkService	PertemuanPembelajaranDm	TglSmtModul	
	SatManService			
	TglSmtService			
KrsController	AnggotaRombelService	KrsDm	AnggotaRombelModul	Krs Transaction
	KrsService	TglSmtDm	KrsModul	
	PdService	PdDm	PdModul	
	PembService	PembDm	PembModul	
	TglSmtService	AnggotaRombelDm	TglSmtModul	
Manajemen KrsController	AturanPenggantiService	AturanPenggantiDm	AturanPenggantiModul	Manajemen Krs Transaction
	BatasAmbilSksService	TglSmtDm	BatasAmbilSksModul	
	IpkService	ThnAjaranDm	IpkModul	
	IpsService	SmtDm	IpsModul	

Nama Controller	Service Layer	Domain Model	Table Module	Transaction Script
	KrsService	PembDm	KrsModul	
	KurikulumService	PdDm	PdModul	
	ManajemenKrs Service	ManajemenKrs Dm	PembModul	
	MKService	AturanPengganti Dm	Pembayaran Modul	
	PdService	IpsDm	PeranPengguna Modul	
	Pembayaran Service	IpkDm	PrasyaratMK Modul	
	PembService	BatasAmbilSks Dm	SatManMK Modul	
	PendidikPengajar Seservice	PembayaranDm	SmtModul	
	PeranPengguna Service	PrasyaratMKDm	TglSmtModul	
	PrasyaratMK Service	SatManMKDm	ThnAjaran Modul	
	PtkService	PeranPengguna Dm		
	SatManMK Service			
	SatManService			
	SmtService			
	TglSmtService			
	ThnAjaranService			
PdController	PdService	PdDm	PdModul	Pd Transaction
	PtkService	PtkDm	PtkModul	
Pembayaran Controller	Pembayaran Service	PembayaranDm	PdModul	Pembayaran Transaction
	SatManService	ThnAjaranDm	Pembayaran Modul	
	SmtService	SmtDm	SatManModul	
	TglService	SatManDm	SmtModul	
	ThnAjaranService	TglSmtDm	TglSmtModul	
		PdDm	ThnAjaran	

Nama Controller	Service Layer	Domain Model	Table Module	Transaction Script
			Modul	
Pemb Controller	KrsService	PembDm	KrsModul	Pemb Transaction
	KurikulumService	MKDm	Kurikulum Modul	
	MkService	TglSmtDm	MKModul	
	PdService	KurikulumDm	PdModul	
	PembSatMan Service	PendidikPengajar Dm	PembModul	
	PembService	PtkDm	PembSatMan Modul	
	PendidikPengajar Service	PdDm	PendidikPeng ajarModul	
	PtkService	KrsDm	PtkModul	
	RombelService	SatManDm	RombelModul	
	SatManService	PembSatManDm	SatManModul	
	TglSmtService	RombelDm	TglSmtModul	
Perwalian Controller	PdService	PtkDm	PdModul	Perwalian Transaction
	PtkService	PdDm	PtkModul	
PtkController	PtkService	PtkDm	PtkModul	Ptk Transaction
	PdService	PdDm		
Rombel Controller	RombelService	RombelDm	Anggota RombelModul	Rombel Transaction
	AnggotaRombel Service	AnggotaRombel Dm	PdModul	
	PdService	PdDm	RombelModul	
	SatManService	SatManDm	SatManModul	
SmtController	SmtService	SmtDm	SmtModul	Smt Transaction
StsKehadiran Controller	StsKehadiran Service	StsKehadiranDm	StsKehadiran Modul	StsKehadira nTransaction
TglSmt Controller	TglSmtService	TglSmtDm	SmtModul	TglSmt Transaction
	ThnAjaranService	ThnAjaranDm	TglSmtModul	
	SmtService	SmtDm	ThnAjaran	

Nama Controller	Service Layer	Domain Model	Table Module	Transaction Script
			Modul	
ThnAjaranController	ThnAjaranService	ThnAjaranDm	ThnAjaran Modul	ThnAjaran Transaction

Selain dependency dari controller, penerapan pola perancangan juga berpengaruh dengan kompleksitas program. Table 4.6 menunjukan perbandingan kompleksitas program

Tabel 4.6. Perbandingan Kompleksitas Tiap Pola Perancangan

Faktor	Service Layer	Transaction Script	Domain Model	Table Module
Jumlah Kelas	147	57	87	85
Line of Code	5522	5878	6073	5591
Jumlah Method	726	454	576	450
Total Cyclomatic	1427	1331	1310	1141

BAB V

PENGUJIAN DAN EVALUASI

5.1. Pengujian Fitur-fitur pada Sistem Baru

Setelah menerapkan tiga pola perancangan baru pada sistem, langkah selanjutnya adalah melakukan pengukuran kualitas dari sistem. Namun sebelum melakukan pengukuran kualitas dari sistem kita harus memastikan bahwa semua fitur-fitur pada sistem baru yang telah dibuat dapat berjalan dengan baik sesuai dengan sistem awal. Untuk pengujian akan dilakukan sesuai dengan skenario uji pada Lampiran B. Berikut hasil pengujian dari ketiga pola perancangan

Tabel 5.1. Hasil Pengujian Fungsi-Fungsi pada Penerapan Tiga Pola Perancangan Baru

No	Uji	Pola Perancangan		
		<i>Domain Logic</i>	<i>Table Module</i>	<i>Transaction Script</i>
1	Menguji fungsi tambah tahun ajaran.	√	√	√
2	Menguji fungsi sunting tahun ajaran.	√	√	√
3	Menguji fungsi hapus tahun ajaran.	√	√	√
4	Menguji fungsi tambah semester.	√	√	√
5	Menguji fungsi sunting semester.	√	√	√
6	Menguji fungsi hapus semester.	√	√	√
7	Menguji fungsi tambah batas pengambilan SKS.	√	√	√
8	Menguji fungsi sunting batas pengambilan SKS.	√	√	√
9	Menguji fungsi hapus batas pengambilan SKS.	√	√	√

No	Uji	Pola Perancangan		
		<i>Domain Logic</i>	<i>Table Module</i>	<i>Transaction Script</i>
10	Menguji fungsi tambah status absensi.	√	√	√
11	Menguji fungsi sunting status absensi.	√	√	√
12	Menguji fungsi hapus status absensi.	√	√	√
13	Menguji fungsi tambah periode.	√	√	√
14	Menguji fungsi sunting periode.	√	√	√
15	Menguji fungsi hapus periode.	√	√	√
16	Menguji fungsi tambah pembelajaran.	√	√	√
17	Menguji fungsi sunting pembelajaran.	√	√	√
18	Menguji fungsi hapus pembelajaran.	√	√	√
19	Menguji fungsi tambah pengajar.	√	√	√
20	Menguji fungsi hapus pengajar.	√	√	√
21	Menguji fungsi tambah peserta.	√	√	√
22	Menguji fungsi tambah peserta dari rombel.	√	√	√
23	Menguji fungsi hapus peserta.	√	√	√
24	Menguji fungsi tambah rombongan belajar.	√	√	√
25	Menguji fungsi sunting rombongan belajar.	√	√	√
26	Menguji fungsi hapus rombongan belajar.	√	√	√
27	Menguji fungsi tambah anggota.	√	√	√
28	Menguji fungsi hapus anggota.	√	√	√

No	Uji	Pola Perancangan		
		<i>Domain Logic</i>	<i>Table Module</i>	<i>Transaction Script</i>
29	Menguji fungsi tambah anak wali.	√	√	√
30	Menguji fungsi lepasanak wali.	√	√	√
31	Menguji fungsi absensi peserta didik.	√	√	√
32	Menguji fungsi absensi pendidik.	√	√	√
33	Menguji fungsi mengambil pembelajaran.	√	√	√
34	Menguji fungsi menghapus pembelajaran.	√	√	√
35	Menguji fungsi menambah pembelajaran.	√	√	√
36	Menguji fungsi rekap absensi peserta didik.	√	√	√
37	Menguji fungsi rekap absensi pendidik.	√	√	√
38	Menguji fungsi rekap berita acara.	√	√	√
39	Menguji fungsi persetujuan RKS.	√	√	√
40	Menguji fungsi membatalkan persetujuan.	√	√	√
41	Menguji fungsi tambah berita acara.	√	√	√
42	Menguji fungsi rekap pembayaran peserta didik.	√	√	√
43	Menguji fungsi sunting berita acara.	√	√	√
44	Menguji aturan pengambilan matakulliah ITS	√	√	√
45	Menguji aturan pengambilan matakulliah PENS	√	√	√

No	Uji	Pola Perancangan		
		<i>Domain Logic</i>	<i>Table Module</i>	<i>Transaction Script</i>
46	Menguji autan pengambilan matakulliah UPN	√	√	√
47	Menguji perubahan kebijakan.	√	√	√

*keterangan:

√ = berhasil

✗ = gagal

Dari tabel 5.1 didapat bahwa tiga sistem baru yang menerapkan tiga pola perancangan baru dapat berjalan sebagai mana mestinya. Setelah pengujian terhadap sistem baru berhasil, maka sistem baru sudah dapat dilakukan pengukuran kualitas.

5.2. Uji Kualitas Pola Perancangan

Pengujian kualitas sistem informasi akademik ini dilakukan untuk tiap-tiap pola perancangan. Pengukuran kualitas dilakukan sesuai dengan deskripsi kualitas pada tabel 3.57-3.66. Tujuan dari kegiatan pengukuran kualitas adalah mendapatkan nilai kualitas X. sebelum mendapatkan nilai dari X dibutuhkan nilai-nilai variable a dan b yang akan dihitung menggunakan formula. Nilai a dan b sendiri dapat dicari dengan bantuan nilai penting seperti yang telah ditentukan.

Pada bab 5.2 ini akan ditampilkan data-data hasil pengukuran tiap poin kualitas dan tiap pola perancangan secara garis besar (nilai a, b dan hasil) namun sebelum itu akan ditentukan dahulu kelompok uji pada sub bab 5.2.1 sebagai pedoman pengujian .

5.2.1. Kelompok Uji

Pada pengujian akan ada bagian uji yang menunjukan bagian mana dari program yang akan dilakukan pengujian. Ada empat kelompok uji yang akan menjadi pedoman pengujian. Kelompok-kelompok tersebut adalah

1. Kelompok Fungsi atau fitur program

Fitur program ini terdiri dari 15 item, yaitu

Tabel 5.2. Kelompok Fungsi atau Fitur Program

No	Deskripsi
1	Mengelola tahun ajaran
2	Mengelola semester
3	Mengelola periode
4	Mengelola batas pengambilan sks
5	Mengelola status absensi
6	Mengelola pembelajaran
7	Mengelola pendamping akademik
8	Mengelola Rombongan belajar
9	Mengelola absensi
10	Menyusun KRS
11	Mencetak KRS
12	Menyetujui KRS
13	Laporan pertemuan
14	Laporan pembayaran
15	Mengelola berita acara

2. Kelompok Interface

Interface digunakan untuk melakukan pengujian yang melakukan pengecekan terhadap penggunaan ulang komponen karena kelompok interface ini yang dipanggil saat diperlukan untuk melakukan proses. Kelompok interface pada program ini berisi 170 kelas pada table 5.3

Tabel 5.3. Daftar Kelompok Interface

No	Bagian di Uji	Paket
1	AbsensiController	Controller (semua pola perancangan)
2	AturanPenggantiController	
3	BatasAmbilSksController	

No	Bagian di Uji	Paket
4	BeritaAcaraController	
5	KrsController	
6	ManajemenKrsController	
7	PdController	
8	PembayaranController	
9	PembController	
10	PerwalianController	
11	PtkController	
12	RombelController	
13	SmtController	
14	StsKehadiranController	
15	TglSemesterController	
16	ThnAjaranController	
17	AnggotaRombelRepository	Repository (Pola perancangan Service Layer)
18	AturanPenggantiRepository	
19	BatasAmbilSksRepository	
20	IpkRepository	
21	IpsRepository	
22	KrsRepository	
23	KurikulumRepository	
24	MenuPeranRepository	
25	MKRepository	
26	PdRepository	
27	PembayaranRepository	
28	PembRepository	
29	PembSatManRepository	
30	PendidikPengajarRepository	

No	Bagian di Uji	Paket
31	PenggunaRepository	
32	PeranPenggunaRepository	
33	PeranRepository	
34	PertemuanPembelajaranRepository	
35	PrasyaratMKRepository	
36	PresensiPdRepository	
37	PresensiPengajarRepository	
38	PtkRepository	
39	RombelRepository	
40	SatManMKRepository	
41	SatManRepository	
42	SmtRepository	
43	StatusKuisisionerRepository	
44	StsKehadiranRepository	
45	TglSmtRepository	
46	ThnAjaranRepository	
47	AbsensiPendidik	Service (Pola perancangan Service Layer)
48	AbsensiPesertaDidik	
49	AjaxResponse	
50	AnggotaRombelService	
51	AturanPenggantiService	
52	BatasAmbilSksService	
53	Datatable	
54	DatatableExtractParam	
55	IpkService	
56	IpsService	
57	KrsReport	

No	Bagian di Uji	Paket
58	KrsService	
59	KrsTransaction	
60	KurikulumService	
61	MajajemenKrsService	
62	MenuPeranService	
63	MKServiceService	
64	PdService	
65	PembayaranService	
66	PembSatManService	
67	PembService	
68	PendidikPengajarService	
69	PenggunaService	
70	PeranPenggunaService	
71	PeranService	
72	PertemuanPembelajaranService	
73	PrasyaratMKService	
74	PresensiPdService	
75	PresensiPengajarService	
76	PtkService	
77	RombelService	
78	SatManMKService	
79	SatManService	
80	SmtService	
81	StatusKuisisionerService	
82	StsKehadiranService	
83	TglSmtService	
84	ThnAjaranService	

No	Bagian di Uji	Paket
85	TglSmtValidator	Validator (Semua pola perancangan)
86	AturanPengganti	
87	AbsensiPendidik.java	Data (Pola perancangan Domain Model, Transaction Script dan Table Module)
88	AbsensiPesertadidik.java	
89	AjaxResponse.java	
90	Datatable.java	
91	DatatableExtractParams.java	
92	KrsReport.java	
93	KrsTransaction.java	
94	AnggotaRombelDm.java	Domain (Pola perancangan Domain Model)
95	AturanPenggantiDm.java	
96	BatasAmbilSksDm.java	
97	IpkDm.java	
98	IpsDm.java	
99	KrsDm.java	
100	KurikulumDm.java	
101	ManajemenKrsDm.java	
102	MenuPeranDm.java	
103	MKDm.java	
104	PdDm.java	
105	PembayaranDm.java	
106	PembDm.java	
107	PembSatManDm.java	
108	PendidikPengajarDm.java	
109	PenggunaDm.java	
110	PeranDm.java	
111	PeranPenggunaDm.java	

No	Bagian di Uji	Paket
112	PertemuanPembelajaranDm.java	
113	PrasyaratMKDm.java	
114	PresensiPdDm.java	
115	PresensiPengajarDm.java	
116	PtkDm.java	
117	RombelDm.java	
118	SatManDm.java	
119	SatManMKDm.java	
120	SmtDm.java	
121	StatusKuisisionerDm.java	
122	StsKehadiranDm.java	
123	TglSmtDm.java	
124	ThnAjaranDm.java	
125	AnggotaRombelModul.java	Modul (Pola perancangan Table Module)
126	AturanPenggantiModul.java	
127	BatasAmbilSksModul.java	
128	IpkModul.java	
129	IpsModul.java	
130	KrsModul.java	
131	KurikulumModul.java	
132	MenuPeranModul.java	
133	MKModul.java	
134	PdModul.java	
135	PembayaranModul.java	
136	PembModul.java	
137	PembSatManModul.java	
138	PendidikPengajarModul.java	

No	Bagian di Uji	Paket
139	PenggunaModul.java	
140	PeranModul.java	
141	PeranPenggunaModul.java	
142	PertemuanPembelajaranModul.java	
143	PrasyaratMKModul.java	
144	PresensiPdModul.java	
145	PresensiPengajarModul.java	
146	PtkModul.java	
147	RombelModul.java	
148	SatManMKModul.java	
149	SatManModul.java	
150	SmtModul.java	
151	StatusKuisisionerModul.java	
152	StsKehadiranModul.java	
153	TglSmtModul.java	
154	ThnAjaranModul.java	Transaction (Pola perancangan <i>Transaction Script</i>)
155	AbsensiTransaction.java	
156	AturanPenggantiTransaction.java	
157	BatasAmbilSksTransaction.java	
158	BeritaAcaraTransaction.java	
159	KrsTransaction.java	
160	ManajemenKrsTransaction.java	
161	PdTransaction.java	
162	PembayaranTransaction.java	
163	PembTransaction.java	
164	PerwalianTransaction.java	
165	PtkTransaction.java	

No	Bagian di Uji	Paket
166	RombelTransaction.java	
167	SmtTransaction.java	
168	StsKehadiranTransaction.java	
169	TglSmtTransaction.java	
170	ThnAjaranTransaction.java	

3. Kelompok Kelas Implementasi

Kelompok kelas implementasi digunakan untuk melakukan pengujian yang melakukan pengecekan terhadap source code komponen karena kelompok implementasi ini yang berisi source code sistem. Kelompok Implementasi pada program ini berisi 170 kelas pada table 5.4

Tabel 5.4. Daftar Kelompok Implementasi

No	Bagian di Uji	Paket
1	AbsensiController	Controller (Semua Pola perancangan)
2	AturanPenggantiController	
3	BatasAmbilSksController	
4	BeritaAcaraController	
5	KrsController	
6	ManajemenKrsController	
7	PdController	
8	PembayaranController	
9	PembController	
10	PerwalianController	
11	PtkController	
12	RombelController	
13	SmtController	
14	StsKehadiranController	

No	Bagian di Uji	Paket
15	TglSemesterController	
16	ThnAjaranController	
17	AnggotaRombelRepositoryImpl	Repository (Pola perancangan Service Layer)
18	AturanPenggantiRepositoryImpl	
19	BatasAmbilSksRepositoryImpl	
20	IpkRepositoryImpl	
21	IpsRepositoryImpl	
22	KrsRepositoryImpl	
23	KurikulumRepositoryImpl	
24	MenuPeranRepositoryImpl	
25	MKRepositoryImpl	
26	PdRepositoryImpl	
27	PembayaranRepositoryImpl	
28	PembRepositoryImpl	
29	PembSatManRepositoryImpl	
30	PendidikPengajarRepositoryImpl	
31	PenggunaRepositoryImpl	
32	PeranPenggunaRepositoryImpl	
33	PeranRepositoryImpl	
34	PertemuanPembelajaranRepository	
35	PrasyaratMKRepositoryImpl	
36	PresensiPdRepositoryImpl	
37	PresensiPengajarRepositoryImpl	
38	PtkRepositoryImpl	
39	RombelRepositoryImpl	
40	SatManMKRepositoryImpl	
41	SatManRepositoryImpl	

No	Bagian di Uji	Paket
42	SmtRepositoryImpl	
43	StatusKuisisionerRepositoryImpl	
44	StsKehadiranRepositoryImpl	
45	TglSmtRepositoryImpl	
46	ThnAjaranRepositoryImpl	
47	AbsensiPendidik	Service (Pola perancangan Service Layer)
48	AbsensiPesertaDidik	
49	AjaxResponse	
50	AnggotaRombelServiceImpl	
51	AturanPenggantiServiceImpl	
52	BatasAmbilSksServiceImpl	
53	Datatable	
54	DatatableExtractParam	
55	IpkServiceImpl	
56	IpsServiceImpl	
57	KrsReport	
58	KrsServiceImpl	
59	KrsTransaction	
60	KurikulumServiceImpl	
61	MajajemenKrsServiceImpl	
62	MenuPeranServiceImpl	
63	MKServiceServiceImpl	
64	PdServiceImpl	
65	PembayaranServiceImpl	
66	PembSatManServiceImpl	
67	PembServiceImpl	
68	PendidikPengajarServiceImpl	

No	Bagian di Uji	Paket
69	PenggunaServiceImpl	
70	PeranServiceImpl	
71	PeranServiceImpl	
72	PertemuanPembelajaranServiceImpl	
73	PrasyaratMKServiceImpl	
74	PresensiPdServiceImpl	
75	PresensiPengajarServiceImpl	
76	PtkServiceImpl	
77	RombelServiceImpl	
78	SatManMKServiceImpl	
79	SatManServiceImpl	
80	SmtServiceImpl	
81	StatusKuisisionerServiceImpl	
82	StsKehadiranServiceImpl	
83	TglSmtServiceImpl	
84	ThnAjaranServiceImpl	
85	TglSmtValidator	Validator (Semua Pola perancangan)
86	AturanPengganti	
87	AbsensiPendidik.java	Data (Pola perancangan Transaction Scipt, Domain Model dan Table Module)
88	AbsensiPesertadidik.java	
89	AjaxResponse.java	
90	Datatable.java	
91	DatatableExtractParams.java	
92	KrsReport.java	
93	KrsTransaction.java	
94	AnggotaRombelDmImpl.java	Domain

No	Bagian di Uji	Paket
95	AturanPenggantiDmImpl.java	(Pola perancangan Domain Model)
96	BatasAmbilSksDmImpl.java	
97	IpkDmImpl.java	
98	IpsDmImpl.java	
99	KrsDmImpl.java	
100	KurikulumDmImpl.java	
101	ManajemenKrsDmImpl.java	
102	MenuPeranDmImpl.java	
103	MKDmImpl.java	
104	PdDmImpl.java	
105	PembayaranDmImpl.java	
106	PembDmImpl.java	
107	PembSatManDmImpl.java	
108	PendidikPengajarDmImpl.java	
109	PenggunaDmImpl.java	
110	PeranDmImpl.java	
111	PeranPenggunaDmImpl.java	
112	PertemuanPembelajaranDmImpl.java	
113	PrasyaratMKDmImpl.java	
114	PresensiPdDmImpl.java	
115	PresensiPengajarDmImpl.java	
116	PtkDmImpl.java	
117	RombelDmImpl.java	
118	SatManDmImpl.java	
119	SatManMKDmImpl.java	
120	SmtDmImpl.java	
121	StatusKuisionerDmImpl.java	

No	Bagian di Uji	Paket
122	StsKehadiranDmImpl.java	
123	TglSmtDmImpl.java	
124	ThnAjaranDmImpl.java	
125	AnggotaRombelModulImpl.java	Modul (Pola perancangan <i>Table Module</i>)
126	AturanPenggantiModulImpl.java	
127	BatasAmbilSksModulImpl.java	
128	IpkModulImpl.java	
129	IpsModulImpl.java	
130	KrsModulImpl.java	
131	KurikulumModulImpl.java	
132	MenuPeranModulImpl.java	
133	MKModulImpl.java	
134	PdModulImpl.java	
135	PembayaranModulImpl.java	
136	PembModulImpl.java	
137	PembSatManModulImpl.java	
138	PendidikPengajarModulImpl.java	
139	PenggunaModulImpl.java	
140	PeranModulImpl.java	
141	PeranPenggunaModulImpl.java	
142	PertemuanPembelajaranModulImpl.java	
143	PrasyaratMKModulImpl.java	
144	PresensiPdModulImpl.java	
145	PresensiPengajarModulImpl.java	
146	PtkModulImpl.java	
147	RombelModulImpl.java	
148	SatManMKModulImpl.java	

No	Bagian di Uji	Paket
149	SatManModulImpl.java	
150	SmtModulImpl.java	
151	StatusKuisisionerModulImpl.java	
152	StsKehadiranModulImpl.java	
153	TglSmtModulImpl.java	
154	ThnAjaranModulImpl.java	
155	AbsensiTransactionImpl.java	Transaction (Pola perancangan <i>Transaction Script</i>)
156	AturanPenggantiTransactionImpl.java	
157	BatasAmbilSksTransactionImpl.java	
158	BeritaAcaraTransactionImpl.java	
159	KrsTransactionImpl.java	
160	ManajemenKrsTransactionImpl.java	
161	PdTransactionImpl.java	
162	PembayaranTransactionImpl.java	
163	PembTransactionImpl.java	
164	PerwalianTransactionImpl.java	
165	PtkTransactionImpl.java	
166	RombelTransactionImpl.java	
167	SmtTransactionImpl.java	
168	StsKehadiranTransactionImpl.java	
169	TglSmtTransactionImpl.java	
170	ThnAjaranTransactionImpl.java	

4. Kelompok Uji Coba

Kelompok Uji Coba digunakan untuk melakukan pengujian yang melakukan pengecekan terhadap uji coba system. Data pada kelompok uji coba ini didapat dari laporan penelitian sebelumnya. Kelompok ini berisi 47 skenario pada table 5.5

Tabel 5.5. Daftar Kelompok Uji Coba

No	Skenario
1	Menguji fungsi tambah tahun ajaran.
2	Menguji fungsi sunting tahun ajaran.
3	Menguji fungsi hapus tahun ajaran.
4	Menguji fungsi tambah semester.
5	Menguji fungsi sunting semester.
6	Menguji fungsi hapus semester.
7	Menguji fungsi tambah batas pengambilan SKS.
8	Menguji fungsi sunting batas pengambilan SKS.
9	Menguji fungsi hapus batas pengambilan SKS.
10	Menguji fungsi tambah status absensi.
11	Menguji fungsi sunting status absensi.
12	Menguji fungsi hapus status absensi.
13	Menguji fungsi tambah periode.
14	Menguji fungsi sunting periode.
15	Menguji fungsi hapus periode.
16	Menguji fungsi tambah pembelajaran.
17	Menguji fungsi sunting pembelajaran.
18	Menguji fungsi hapus pembelajaran.
19	Menguji fungsi tambah pengajar.
20	Menguji fungsi hapus pengajar.
21	Menguji fungsi tambah peserta.
22	Menguji fungsi tambah peserta dari rombel.
23	Menguji fungsi hapus peserta.
24	Menguji fungsi tambah rombongan belajar.
25	Menguji fungsi sunting rombongan belajar.
26	Menguji fungsi hapus rombongan belajar.

No	Skenario
27	Menguji fungsi tambah anggota.
28	Menguji fungsi hapus anggota.
29	Menguji fungsi tambah anak wali.
30	Menguji fungsi lepasanak wali.
31	Menguji fungsi absensi peserta didik.
32	Menguji fungsi absensi pendidik.
33	Menguji fungsi mengambil pembelajaran.
34	Menguji fungsi menghapus pembelajaran.
35	Menguji fungsi menambah pembelajaran.
36	Menguji fungsi rekap absensi peserta didik.
37	Menguji fungsi rekap absensi pendidik.
38	Menguji fungsi rekap berita acara.
39	Menguji fungsi persetujuan RKS.
40	Menguji fungsi membatalkan persetujuan.
41	Menguji fungsi tambah berita acara.
42	Menguji fungsi rekap pembayaran peserta didik.
43	Menguji fungsi sunting berita acara.
44	Menguji aturan pengambilan matakulliah ITS
45	Menguji aturan pengambilan matakulliah PENS
46	Menguji aturan pengambilan matakulliah UPN
47	Menguji perubahan kebijakan.

5.2.2. Uji Kualitas Pola Perancangan *Service Layer*

Pada sub bab ini akan menyajikan hasil pengukuran kualitas sistem yang dibangun dengan pola perancangan *Service Layer*. Pengukuran kualitas dilakukan pada 10 poin kualitas. Hasil dari pengukuran kualitas disajikan pada Lampiran D sedangkan ringkasan dapat dilihat pada tabel 5.6.

Tabel 5.6. Ringkasa Hasil Pengukuran Kualitas pada *Service Layer*

Quality Sub-Characteristic	Point	Kode	A	B	X
Modularity	coupling of component conformance	mmo-1-G	16	32	0,50
	cyclomatic complexity	mmo-2-S	6	723	0,99
Reusability	Reusability of Assets	mre-1-G	70	86	0,81
	conformance to coding rule	mre-2-S	84	147	0,57
Analysability	system log complateness conformace	man-1-G	5	15	0,33
	diagnosis function effectiveness	man-2-S	15	15	1,00
	diagnosis function suffiency conformance	man-3-S	15	15	1,00
Testability	Test function complateness conformance	mte-1-G	50	50	1,00
	Autonomus testability	mte-2-S	0	50	0,00
	Test Restartbility	mte-3-S	0	50	0,00

5.2.3. Uji Kualitas Pola Perancangan *Transaction Script*

Pada sub bab ini akan menyajikan hasil pengukuran kualitas sistem yang dibangun dengan pola perancangan *Transaction Script*. Pengukuran kualitas dilakukan pada 10 poin kualitas. Hasil dari pengukuran kualitas disajikan pada Lampiran D sedangkan ringkasan dapat dilihat pada tabel 5.7.

**Tabel 5.7. Ringkasan Hasil Pengukuran Kualitas pada
*Transaction Script***

Quality Sub-Characteristic	Point	Kode	A	B	X
Modularity	coupling of component conformance	mmo-1-G	16	25	0,64
	cyclomatic complexity	mmo-2-S	6	454	0,99
Reusability	Reusability of Assets	mre-1-G	9	41	0,22
	conformance to coding rule	mre-2-S	8	57	0,14
Analysability	system log complateness conformance	man-1-G	15	15	1,00
	diagnosis function effectiveness	man-2-S	15	15	1,00
	diagnosis function suffiency conformance	man-3-S	15	15	1,00
Testability	Test function complateness conformance	mte-1-G	50	50	1,00
	Autonomus testability	mte-2-S	0	50	0,00
	Test Restartbility	mte-3-S	0	50	0,00

5.2.4. Uji Kualitas Pola Perancangan *Domain Model*

Pada sub bab ini akan menyajikan hasil pengukuran kualitas sistem yang dibangun dengan pola perancangan *Domain Model*. Pengukuran kualitas dilakukan pada 10 poin kualitas. Hasil dari pengukuran kualitas disajikan pada Lampiran D sedangkan ringkasan dapat dilihat pada tabel 5.8.

**Tabel 5.8. Ringkasan Hasil Pengukuran Kualitas pada
Domain Model**

Quality Sub-Characteristic	Point	Kode	A	B	X
Modularity	coupling of component conformance	mmo-1-G	16	40	0,40
	cyclomatic complexity	mmo-2-S	6	576	0,99
Reusability	Reusability of Assets	mre-1-G	40	56	0,71
	conformance to coding rule	mre-2-S	26	87	0,30
Analysability	system log complateness conformance	man-1-G	15	15	1,00
	diagnosis function effectiveness	man-2-S	15	15	1,00
	diagnosis function suffiency conformance	man-3-S	15	15	1,00
Testability	Test function complateness conformance	mte-1-G	50	50	1,00
	Autonomus testability	mte-2-S	0	50	0,00
	Test Restartbility	mte-3-S	0	50	0,00

5.2.5. Uji Kualitas Pola Perancangan *Table Module*

Pada sub bab ini akan menyajikan hasil pengukuran kualitas sistem yang dibangun dengan pola perancangan *Table Module*. Pengukuran kualitas dilakukan pada 10 poin kualitas. Hasil dari pengukuran kualitas disajikan pada Lampiran D sedangkan ringkasan dapat dilihat pada tabel 5.9.

Tabel 5.9. Ringkasan Hasil Pengukuran Kualitas Test function complateness conformance pada *Table Module*

Quality Sub-Characteristic	Point	Kode	A	B	X
Modularity	coupling of component conformance	mmo-1-G	16	39	0,41
	cyclomatic complexity	mmo-2-S	6	450	0,99
Reusability	Reusability of Assets	mre-1-G	39	55	0,71
	conformance to coding rule	mre-2-S	34	85	0,40
Analysability	system log complateness conformance	man-1-G	15	15	1,00
	diagnosis function effectiveness	man-2-S	15	15	1,00
	diagnosis function sufficiency conformance	man-3-S	15	15	1,00
Testability	Test function complateness conformance	mte-1-G	50	50	1,00
	Autonomus testability	mte-2-S	0	50	0,00
	Test Restartbility	mte-3-S	0	50	0,00

5.3. Perbandingan Hasil Kualitas Sistem

Setelah melakukan pengukuran kualitas pada keempat pola perancangan, didapatkan hasil lengkap pengukuran kualitas yang hasilnya ditampilkan pada table 5.10. Pada tabel 5.11 dapat dilihat perbandingan dari kualitas masing-masing pola perancangan.

Tabel 5.10. Perbandingan Kualitas pada Penerapan Pola Perancangan

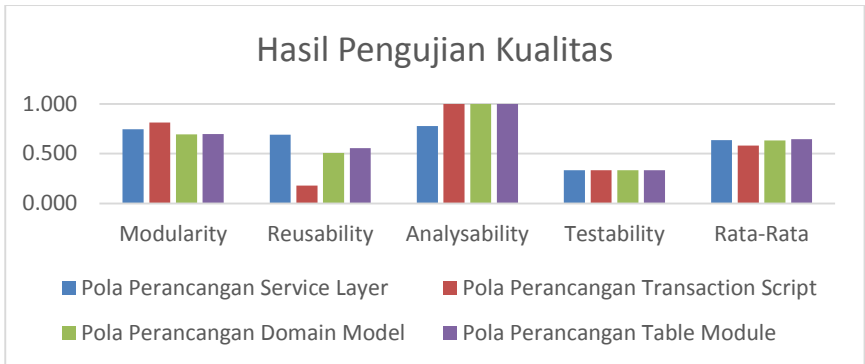
Quality Sub-Characteristic	Point	Kode	Pola Perancangan			
			Service Layer	Transaction Script	Domain Model	Table Module
Modularity	coupling of component conformance	mmo-1-G	0,500	0,640	0,400	0,410
	cyclomatic complexity	mmo-2-S	0,992	0,987	0,990	0,987
	Rata-rata		0,746	0,813	0,695	0,698
Reusability	Reusability of Assets	mre-1-G	0,814	0,220	0,714	0,709
	conformance to coding rule	mre-2-S	0,571	0,140	0,299	0,400
	Rata-rata		0,693	0,180	0,507	0,555
Analysability	system log complateness conformace	man-1-G	0,333	1,000	1,000	1,000
	diagnosis function effectiveness	man-2-S	1,000	1,000	1,000	1,000
	diagnosis function suffiency conformance	man-3-S	1,000	1,000	1,000	1,000
	Rata-rata		0,778	1,000	1,000	1,000
Testability	Test function complateness conformance	mte-1-G	1,000	1,000	1,000	1,000
	Autonomus testability	mte-2-S	0,000	0,000	0,000	0,000
	Test Restartbility	mte-3-S	0,000	0,000	0,000	0,000
	Rata-rata		0,333	0,333	0,333	0,333
Rata-rata Kualitas			0,637	0,582	0,634	0,647

Tabel 5.11 dan gambar 5.1 menyajikan hasil kualitas dari masing-masing sub-karakteristik kualitas. Dari tabel dan grafik dapat disimpulkan bahwa penerapan pola perancangan yang berbeda memiliki pengaruh yang berbeda-beda pada nilai sub-karakteristik. Pengertian dari pengaruh yang berbeda-beda adalah ada sub-karakteristik pada kualitas yang bernilai dinamis pada penerapan pola perancangan yang berbeda. Namun juga ada sub-karakteristik yang bernilai statis walaupun telah menerapkan pola perancangan yang berbeda. Pembahasan tentang pengaruh pola perancangan dan sub-kualitas akan dibahas pada bab selanjutnya. Jika membandingkan hasil pengukuran didapatkan kualitas Maintainability terbaik adalah dengan pola perancangan Table Module.

Pemilihan Table Module sebagai pola perancangan domain logic terbaik adalah pemilihan berdasarkan nilai Maintainability secara umum. Namun jika dilihat lebih spesifik, sistem informasi akademik memiliki karakteristik khusus yaitu perubahan fitur yang cukup dinamis dengan artian fitur-fitur akan sering bertambah, berkurang atau mungkin berubah. Dengan karakteristik khusus sistem informasi akademik maka sistem informasi akademik lebih mementingkan sub-kualitas Modularity. Jika melihan dari sub-kualitas Modularity, penerapan pola perancangan paling tepat adalah Transaction Script karena memiliki nilai sub-kualitas Modularity terbaik dengan 0.813 poin.

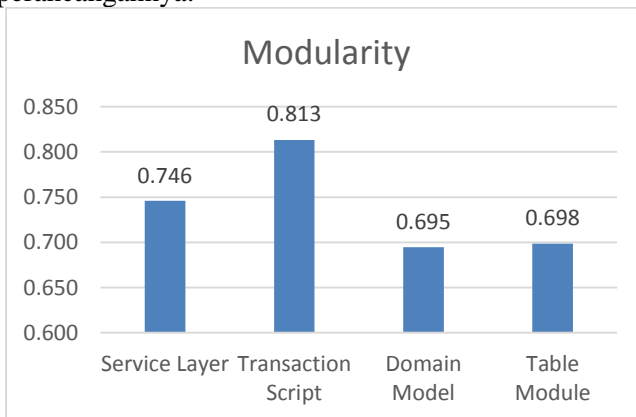
Tabel 5.11. Ringkasan Perbandingan Kualitas Pola Perancangan

Quality Sub-Characteristic	Pola Perancangan			
	<i>Service Layer</i>	<i>Transaction Script</i>	<i>Domain Model</i>	<i>Table Module</i>
Modularity	0,746	0,813	0,695	0,698
Reusability	0,693	0,180	0,507	0,555
Analysability	0,778	1,000	1,000	1,000
Testability	0,333	0,333	0,333	0,333
Rata-Rata	0,637	0,582	0,634	0,647



Gambar 5.1. Grafik Hasil Pengujian Kualitas Maintainability
5.3.1. Sub-Kualitas Modularity

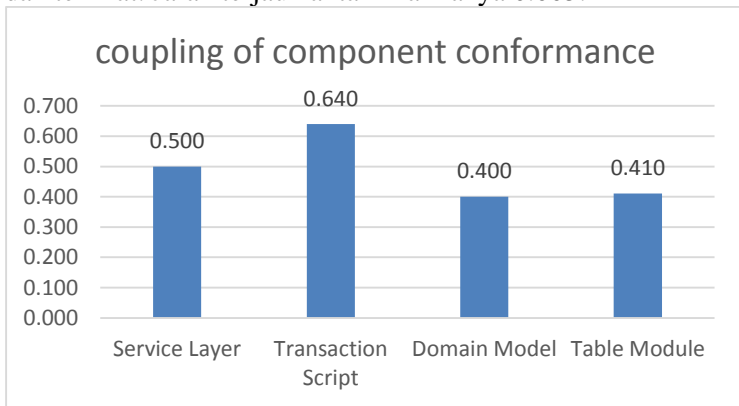
Sub Kualitas Modularity pada kualitas Maintainability ini memiliki dua buah poin penilaian yaitu Coupling of Component conformance dan Cyclomatic complexity. Jika dilihat dari gambar grafik 5.2, sub kualitas modularity adalah sub kualitas yang dipengaruhi pola perancangan. Hal ini bisa dilihat dari nilai sub-kualitas modularity yang bernilai dinamis atau berbeda-beda tiap pola perancangannya.



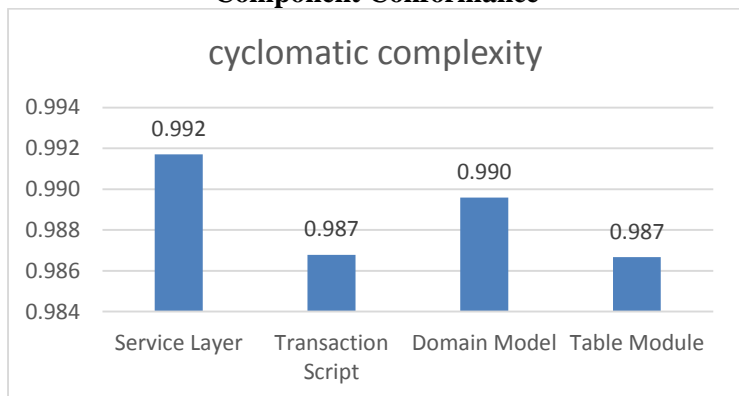
Gambar 5.2. Grafik Hasil Pengujian Sub-Kualitas Modularity

Nilai dinamis dari sub-kualitas modularity juga dipengaruhi dua poin penilaiannya. Jika dilihat dari tabel X, nilai

Coupling of Component conformance dan Cyclomatic complexity juga berbeda-beda tiap pola perancangannya. Walaupun nilai sub-kualitas modularity ini berubah-ubah tiap pola perancangannya, namun perbedaan nilai sangat terlihat pada poin Coupling of Component yang jarak terjauh dari nilainya adalah 0.240. Tidak seperti Coupling of Component, poin kedua pada sub-karakteristik ini yaitu Cyclomatic Complexity perbedaan yang terjadi nyaris tidak terlihat. Jarak terjauh antar nilai hanya 0.005.



Gambar 5.3. Grafik Hasil Pengujian Poin Coupling of Component Conformance



Gambar 5.4. Grafik Hasil Pengujian Poin Cyclomatic Complexity

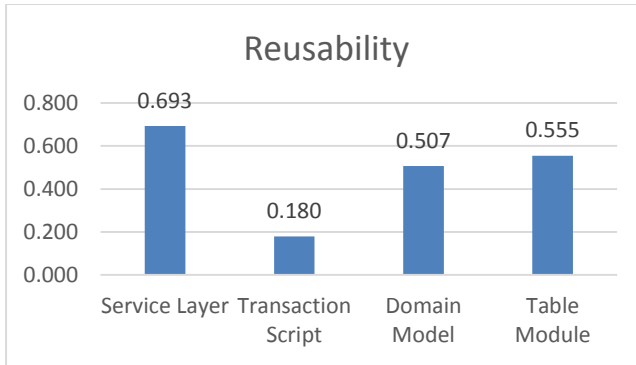
Dari gambar grafik 5.2 dapat ditentukan bahwa pola perancangan terbaik untuk sub-karakteristik Modularity ini adalah pola perancangan *Transaction Script*. Nilai sub-kualitas dari *Transaction Script* juga tidak lepas dari nilai poin Coupling of Component yang cukup tinggi disbanding pola perancangan lain walaupun pada poin cyclomatic mendapatkan nilai terendah.

Jika melihat dari hasil kualitas pada sub-kualitas modularity, nilai sub-kualitas dipengaruhi oleh nilai poin coupling of component dan cyclomatic. Sedangkan nilai poin-poin dipengaruhi oleh nilai penting dari tiap-tiap poin, maka nilai sub-kualitas modularity dipengaruhi oleh nilai-nilai penting dari poin coupling of component dan cyclomatic. Berikut nilai-nilai yang mempengaruhi kualitas modularity beserta hubungannya

1. Jumlah kelas dependent, semakin banyak maka nilai modularity semakin baik.
2. Jumlah kelas yang dependent terhadap suatu kelas, semakin sedikit maka nilai modularity semakin baik.
3. Nilai cyclomatic masing-masing fungsi, semakin kecil nilai cyclomatic maka modularity semakin baik.
4. Jumlah fungsi yang memiliki nilai cyclomatic kurang dari threshold, semakin banyak maka nilai modularity semakin baik.

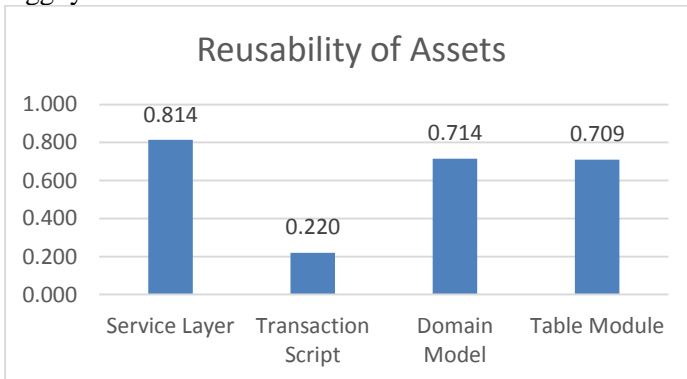
5.3.2. Kualitas Reusability

Sub Kualitas Reusability pada kualitas Maintainability ini memiliki dua buah poin penilaian yaitu Reusability of Assets dan conformance to coding rule. Jika dilihat dari gambar grafik 5.3, sub kualitas reusability adalah sub kualitas yang dipengaruhi pola perancangan. Hal ini bisa dilihat dari nilai sub-kualitas reusability yang bernilai dinamis atau berbeda-beda tiap pola perancangannya.

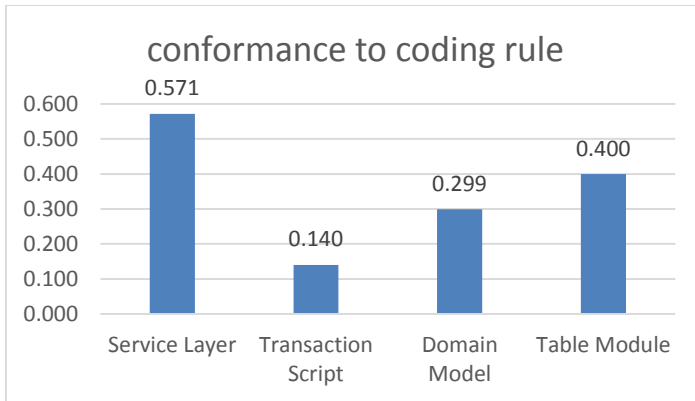


Gambar 5.5. Grafik Hasil Pengukuran Kualitas Reusability

Nilai dinamis dari sub-kualitas reusability juga dipengaruhi dua poin penilaiannya. Jika dilihat dari tabel 5.46, nilai Reusability of Assets dan conformance to coding rule juga berbeda-beda tiap pola perancangannya. Pada poin Reusability of Assets, perbedaan nilai sangat terlihat, jarak nilai tertinggi dan terendah adalah 0.594 (selisih nilai *Service Layer* dengan *Transaction Script*). Untuk poin kedua pada sub-karakteristik ini perbedaan nilai juga cukup lebar dengan nilai perbedaan nilai tertinggi yaitu 0.431.



Gambar 5.6. Grafik Hasil Pengukuran Poin Reusability of Assets



Gambar 5.7. Grafik Hasil Pengukuran Poin Conformance to Coding Rule

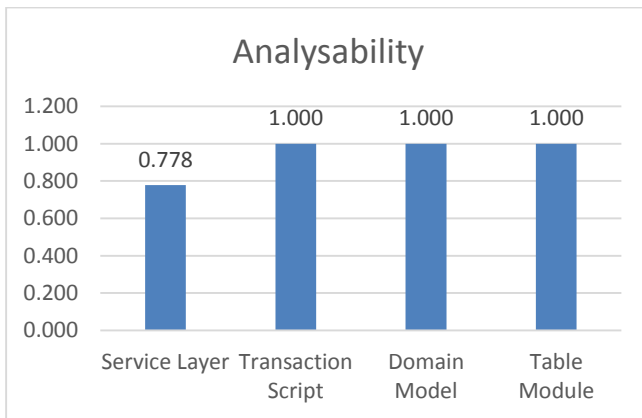
Dari gambar grafik 5.3 dapat ditentukan bahwa pola perancangan terbaik untuk sub-karakteristik Reusability ini adalah pola perancangan *Service Layer*. Nilai sub-kualitas dari *Service Layer* juga tidak lepas dari nilai poin Reusability of Assets dan conformance to coding rule yang cukup tinggi disbanding pola perancangan lain.

Jika melihat dari hasil kualitas pada sub-kualitas reusability, nilai sub-kualitas dipengaruhi oleh nilai poin Reusability of Assets dan conformance to coding rule. Sedangkan nilai poin-poin dipengaruhi oleh nilai penting dari tiap-tiap poin, maka nilai sub-kualitas reusability dipengaruhi oleh nilai-nilai penting dari poin Reusability of Assets dan conformance to coding rule. Berikut nilai-nilai yang mempengaruhi kualitas reusability beserta hubungannya

1. Jumlah kelas yang didesain untuk dapat digunakan ulang, semakin banyak maka nilai reusability semakin baik.
2. Jumlah pemanggilan terhadap kelas, semakin banyak maka nilai reusability semakin baik.
3. Jumlah kelas yang menerapkan aturan pemrograman, semakin banyak maka reusability semakin baik.

5.3.3. Kualitas Analysability

Sub Kualitas Analysability pada kualitas Maintainability ini memiliki tiga buah poin penilaian yaitu system log completeness conformace, diagnosis function effectiveness dan diagnosis function sufficiency conformance. Jika dilihat dari gambar grafik 5.4, sub kualitas analysability adalah sub kualitas yang statis atau tidak dipengaruhi pola perancangan. Hal ini bisa dilihat dari nilai sub-kualitas yang bernilai sama tiap pola perancangannya. Perbedaan nilai hanya terjadi pada *Service Layer*.



Gambar 5.8. Grafik Hasil Pengukuran Sub-Kualitas Analysability

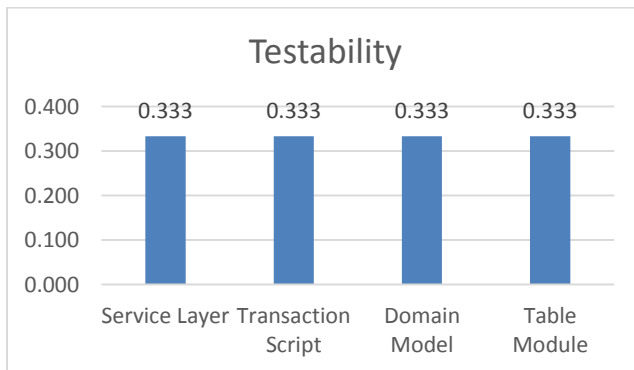
Jika dilihat dari tabel 5.46, nilai tiga poin analyzability semua hampir sama. Perbedaan nilai hanya terdapat pada poin system log completeness pola perancangan *Service Layer*. Pada implementasi nilai berbeda dari system log *Service Layer* bukan dikarenakan penerapan pola perancangan namun karena pada pengembangan sebelumnya system log hanya dibuat pada beberapa fitur sedangkan pada tugas akhir ini sistem lama tidak diubah kecuali untuk memperbaiki bug yang terjadi.

Karena tidak dipengaruhi oleh pola perancangan, sub-karakteristik ini memiliki nilai yang sama untuk semua pola

perancangan *Domain Logic*. Nilai yang sama pada penerapan pola perancangan yang berbeda-beda menyebabkan tidak ada nilai-nilai pada bagian pola perancangan *Domain Logic* yang dapat mempengaruhi nilai analyzability.

5.3.4. Kualitas Testability

Sub Kualitas Testability pada kualitas Maintainability ini memiliki tiga buah poin penilaian yaitu Test function complateness conformance, Autonomus testability dan Test Restartbility. Jika dilihat dari gambar grafik 5.5, sub kualitas Testability adalah sub kualitas yang statis atau tidak dipengaruhi pola perancangan. Hal ini bisa dilihat dari nilai sub-kualitas yang bernilai sama tiap pola perancangannya.



Gambar 5.9. Grafik Hasil Pengukuran Sub-Kualitas Testability

Jika dilihat dari tabel 5.46, nilai tiga poin testability semua sama. Karena tidak dipengaruhi oleh pola perancangan, sub-karakteristik ini memiliki nilai yang sama untuk semua pola perancangan *Domain Logic*. Nilai yang sama pada penerapan pola perancangan yang berbeda-beda menyebabkan tidak ada nilai-nilai pada bagian pola perancangan *Domain Logic* yang dapat mempengaruhi nilai testability.

[Halaman ini Sengaja Dikosongkan]

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut.

6.1. Kesimpulan

Dari proses penerapan pola perancangan dan pengukuran kualitas pada sistem informasi akademik Modul Pembelajaran dapat diambil kesimpulan sebagai berikut:

1. Pola perancangan pada kelompok *Domain Logic* dapat diterapkan pada sistem informasi akademik dengan membuat paket-paket baru sesuai dengan pola perancangan. Paket-paket baru berisi kelas-kelas yang mengelompokkan proses-proses logika sesuai dengan pola perancangannya. Kelas-kelas pola perancangan juga digunakan sebagai pengolah proses yang dibutuhkan oleh controller.
2. Pengukuran kualitas Maintainability pada ISO 25023 dapat dilakukan dengan cara mendefinisikan terlebih dahulu poin-poin penting dari tiap sub kualitas sehingga bisa didapat nilai A dan B pada poin kualitas. Nilai A dan B digunakan untuk mendapatkan nilai X atau hasil kualitas dengan cara memasukkan nilai A dan B ke formula X dari tiap-tiap poin kualitas.
3. Evaluasi terhadap hasil pengukuran dapat dilakukan dengan cara membandingkan hasil nilai kualitas dari masing-masing sub-kualitas dan juga rata-rata dari nilai sub-kualitas.
4. Pengukuran kualitas terhadap empat pola perancangan kelompok *Domain Logic* didapatkan bahwa pola perancangan mempengaruhi kualitas perangkat lunak.

Dari hasil pengukuran kualitas, nilai sub-kualitas modularity terbaik didapatkan pada pola perancangan *Transaction Script* dengan nilai 0,813, sub-kualitas reusability terbaik pada penerapan pola perancangan *Service Layer* dengan nilai 0,693. Sedangkan pada analyzability dan testability tidak dipengaruhi pola perancangan. Secara keseluruhan pengukuran kualitas Maintainability didapat kualitas terbaik adalah *Table Module*.

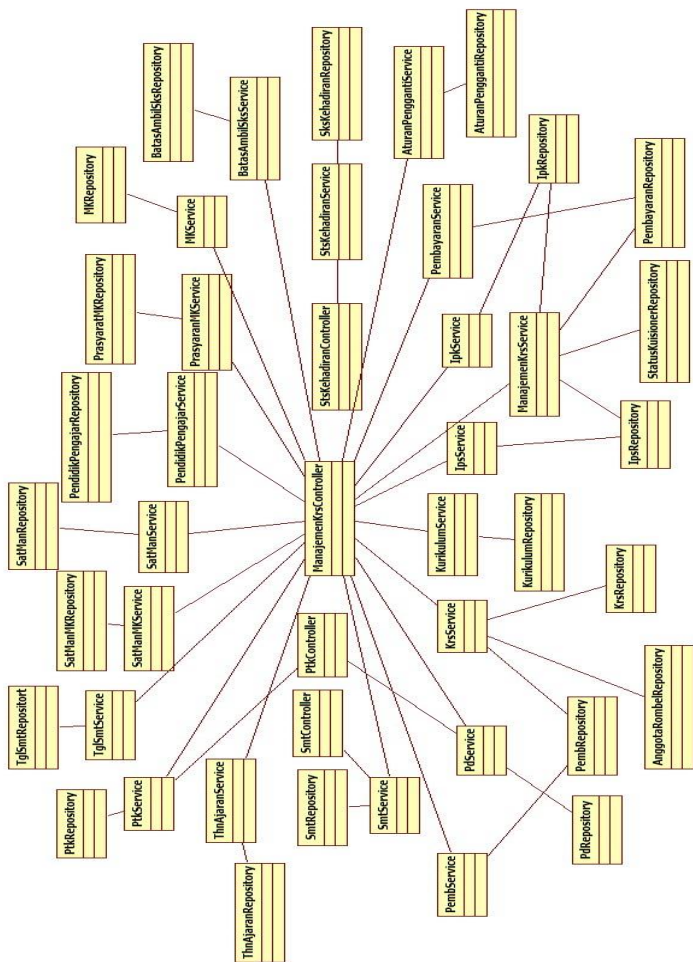
6.2. Saran

Saran yang dapat diberikan pada tugas akhir ini adalah menambahkan kualitas lain selain Maintainability dan menggunakan pola perancangan lain selain kelompok *Domain Logic*. Namun sebelum mencoba menerapkan pola perancangan lain dan kualitas lain harus dilakukan analisis terlebih dahulu tentang hubungan antara kualitas dan pola perancangan.

DAFTAR PUSTAKA

- [1] “ISO 25010.” [Daring]. Tersedia pada: <http://iso25000.com/index.php/en/iso-25000-standards/iso-25010>. [Diakses: 19-Des-2016].
- [2] “ISO/IEC 25023:2016(en), Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Measurement of system and software product quality.” [Daring]. Tersedia pada: <https://www.iso.org/obp/ui/#iso:std:iso-iec:25023:ed-1:v1:en>. [Diakses: 19-Des-2016].
- [3] U. L. Yuhana, G. P. N. Suminto, dan R. N. E. Anggraini, “RANCANG BANGUN COMMERCIAL OFF THE SHELF (COTS) SISTEM INFORMASI AKADEMIK BERBASIS WEB PADA MODUL KELOLA PEMBELAJARAN,” *Jur. Tek. Inform.-ITS*, 2015.
- [4] B. A. Alfirdaus, “Rancang Bangun Perangkat Lunak Sistem Informasi Akademik Berbasis Web dengan Rancangan Modularitas dan Evolusi pada Modul Ekuivalensi,” *Jur. Tek. Inform.-ITS*, 2015.
- [5] S. Rochimah, H. Rahman, dan R. N. E. Anggraini, “RANCANG BANGUN SISTEM INFORMASI AKADEMIK GENERIK PADA MODUL PENILAIAN MENGGUNAKAN POLA PERANCANGAN HIERARCHICAL MODEL-VIEW-CONTROLLER,” *Jur. Tek. Inform.-ITS*, 2015.
- [6] S. Rochimah, R. J. Akbar, dan A. T. Averousi, “RANCANG BANGUN PERANGKAT LUNAK SISTEM INFORMASI AKADEMIK GENERIK PADA MODUL KURIKULUM,” *Jur. Tek. Inform.-ITS*, 2015.
- [7] U. L. Yuhana, R. J. Akbar, dan T. Nurwantoro, “KERANGKA KERJA SINKRONISASI BASIS DATA RELASIONAL BERBASIS WEB PADA STUDI KASUS SISTEM INFORMASI AKADEMIK,” *Jur. Tek. Inform.-ITS*, 2015.

- [8] U. L. Yuhana, R. J. Akbar, dan S. A. Wijaya, "RANCANG BANGUN KERANGKA KERJA SISTEM INFORMASI AKADEMIK MODULAR BERBASIS WEB DENGAN POLA ARSITEKTUR HIERARCHICAL MODEL-VIEW-CONTROLLER," *Jur. Tek. Inform.-ITS*, 2016.
- [9] D. Robert H, *Software Quality: Concepts and Plans*, First. New Jersey: Prentice Hall, 1989.
- [10] I. Richardus Eko, "KRITERIA PENJAMINAN KUALITAS PERANGKAT LUNAK," *Seri 999 E-Artik.*, Nov 2012.
- [11] ~StandardSBoar, "IEEE Standard Glossary of Software Engineering Terminology." The Institute of Electrical and Electronics Engineers 345 East 47th Street, New York, NY 10017, USA, 28-Sep-1990.
- [12] E. Gamma, R. Helm, R. Jahnson, dan J. Vissides, *Design Pattern: Element of Reusable Object-oriented Software*, First. USA: Addison-Wesley, 1994.
- [13] martin fowler, *Patterns of Enterprise Application Architecture*. USA: Addison-Wesley Professional, 2002.



Gambar A.2. Kelas Diagram Sistem Informasi Akademik Versi Replikasi (2)

LAMPIRAN B

Tabel B.1. Detail Pengujian Fungsi Akses Menu

Nama Skenario Pengujian	Uji coba mengakses menu pada aplikasi (uji coba tambahan)
Kode	PF-000
Tujuan Pengujian	Menguji menu-menu yang telah ada di aplikasi
Kondisi Awal	Halaman awal aplikasi
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna login ke aplikasi 2. Pengguna memilih menu-menu yang tersedia sesuai dengan hak aksesnya
Masukan	-
Hasil yang diharapkan	Muncul tampilan sesuai menu yang dipilih
Hasil pengujian	Gagal. Error Kode 500 (servlet error).

Tabel B.2. Detail Pengujian Fungsi Tambah Tahun Ajaran

Nama Skenario Pengujian	Fungsionalitas mengelola tahun ajaran.
Kode	PF-001
Tujuan Pengujian	Menguji fungsi tambah tahun ajaran.
Kondisi Awal	Pengguna merupakan administrator
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu kelola tahun ajaran. 2. Pengguna menekan tombol tambah. 3. Pengguna mengisi formulir tahun ajaran yang belum ada didalam sistem. 4. Pengguna menekan tombol simpan
Masukan	<ol style="list-style-type: none"> 1. Tahun. 2. Minimal pertemuan. 3. Minimal kehadiran peserta didik.

Hasil yang diharapkan	Tahun ajaran baru tersimpan dan notifikasi sukses muncul.
Hasil pengujian	Berhasil

Tabel B.3. Detail Pengujian Fungsi Sunting Tahun Ajaran

Nama Skenario Pengujian	Fungsionalitas mengelola tahun ajaran.
Kode	PF-002
Tujuan Pengujian	Menguji fungsi sunting tahun ajaran.
Kondisi Awal	Pengguna merupakan administrator.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu kelola tahun ajaran. 2. Pengguna menekan tombol <i>edit</i>. 3. Pengguna mengubah formulir tahun ajaran. 4. Pengguna menekan tombol simpan.
Masukan	<ol style="list-style-type: none"> 1. Tahun. 2. Minimal pertemuan. 3. Minimal kehadiran peserta didik.
Hasil yang diharapkan	Tahun ajaran tersunting dan notifikasi sukses muncul
Hasil pengujian	Berhasil

Tabel B.4. Detail Pengujian Fungsi Hapus Tahun Ajaran

Nama Skenario Pengujian	Fungsionalitas mengelola tahun ajaran
Kode	PF-003

Tujuan Pengujian	Menguji fungsi hapus tahun ajaran.
Kondisi Awal	Pengguna merupakan administrator.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu kelola tahun ajaran. 2. Pengguna menekan tombol hapus.
Masukan	
Hasil yang diharapkan	Tahun ajaran terhapus dan notifikasi sukses muncul
Hasil pengujian	Berhasil

Tabel B.5. Detail Pengujian Fungsi Tambah Semester

Nama Skenario Pengujian	Fungsionalitas mengelola semester
Kode	PF-004
Tujuan Pengujian	Menguji fungsi tambah semester.
Kondisi Awal	Pengguna merupakan administrator.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu kelola semester 2. Pengguna menekan tombol tambah 3. Pengguna mengubah formulir tahun ajaran 4. Pengguna menekan tombol simpan
Masukan	<ol style="list-style-type: none"> 1. Nama semester 2. Jumlah minggu 3. Jenis semester

Hasil yang diharapkan	Semester baru tersimpan dan notifikasi sukses muncul.
Hasil pengujian	Berhasil

Tabel B.6.Detail Pengujian Fungsi Sunting Semester

Nama Skenario Pengujian	Fungsionalitas mengelola semester.
Kode	PF-005
Tujuan Pengujian	Menguji fungsi sunting semester.
Kondisi Awal	Pengguna merupakan administrator.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu kelola semester. 2. Pengguna menekan tombol <i>edit</i>. 3. Pengguna mengubah formulir semester. 4. Pengguna menekan tombol simpan.
Masukan	<ol style="list-style-type: none"> 1. Nama semester. 2. Jumlah minggu. 4. Jenis semester.
Hasil yang diharapkan	Semester tersunting dan notifikasi sukses muncul
Hasil pengujian	Berhasil

Tabel B.7.Detail Pengujian Fungsi Hapus Semester

Nama Skenario Pengujian	Fungsionalitas mengelola semester
Kode	PF-006

Tujuan Pengujian	Menguji fungsi hapus semester.
Kondisi Awal	Pengguna merupakan administrator.
Prosedur Pengujian	1. Pengguna membuka menu kelola semester. 2. Pengguna menekan tombol hapus.
Masukan	
Hasil yang diharapkan	Semester terhapus dan notifikasi sukses muncul
Hasil pengujian	Berhasil

**Tabel B.8.Detail Pegujian Fungsi Tambah Batas
Pengambilan SKS**

Nama Skenario Pengujian	Fungsionalitas mengelola batas pengambilan SKS
Kode	PF-007
Tujuan Pengujian	Menguji fungsi tambah batas pengambilan SKS.
Kondisi Awal	Pengguna merupakan administrator.
Prosedur Pengujian	1. Pengguna membuka menu kelola batas pengambilan SKS 2. Pengguna menekan tombol tambah 3. Pengguna mengubah formulir tahun ajaran 4. Pengguna menekan tombol simpan
Masukan	1. Batas bawah IPS 2. Batas pengambilan SKS

Hasil yang diharapkan	Batas pengambilan SKS baru tersimpan dan notifikasi sukses muncul.
Hasil pengujian	Berhasil

Tabel B.9. Detail Pengujian Fungsi Sunting Batas Pengambilan SKS

Nama Skenario Pengujian	Fungsionalitas mengelola batas pengambilan SKS.
Kode	PF-008
Tujuan Pengujian	Menguji fungsi sunting batas pengambilan SKS.
Kondisi Awal	Pengguna merupakan administrator.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu kelola batas pengambilan SKS. 2. Pengguna menekan tombol <i>edit</i>. 3. Pengguna mengubah formulir batas pengambilan SKS. 4. Pengguna menekan tombol simpan.
Masukan	<ol style="list-style-type: none"> 1. Batas bawah IPS 2. Batas pengambilan SKS
Hasil yang diharapkan	Batas pengambilan SKS tersunting dan notifikasi sukses muncul
Hasil pengujian	Berhasil

Tabel B.10.Detail Pengujian Fungsi Hapus Batas Pengambilan SKS

Nama Skenario Pengujian	Fungsionalitas mengelola batas pengambilan SKS
Kode	PF-009
Tujuan Pengujian	Menguji fungsi hapus batas pengambilan SKS.
Kondisi Awal	Pengguna merupakan administrator.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu kelola batas pengambilan SKS. 2. Pengguna menekan tombol hapus.
Masukan	
Hasil yang diharapkan	Batas pengambilan SKS terhapus dan notifikasi sukses muncul
Hasil pengujian	Berhasil

Tabel B.11.Detail Pengujian Fungsi Tambah Status Absensi

Nama Skenario Pengujian	Fungsionalitas mengelola status absensi
Kode	PF-010
Tujuan Pengujian	Menguji fungsi tambah status absensi.
Kondisi Awal	Pengguna merupakan administrator.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu kelola status kehadiran 2. Pengguna menekan tombol tambah

	3. Pengguna mengubah formulir tahun ajaran 4. Pengguna menekan tombol simpan
Masukan	1. Kode status absensi. 2. Nama status absensi. 3. Terhitung absen.
Hasil yang diharapkan	Status absensi baru tersimpan dan notifikasi sukses muncul.
Hasil pengujian	Berhasil

Tabel B.12.Detail Pengujian FUnksi Sunting Status Absesnis

Nama Skenario Pengujian	Fungsionalitas mengelola status absensi.
Kode	PF-011
Tujuan Pengujian	Menguji fungsi sunting status absensi.
Kondisi Awal	Pengguna merupakan administrator.
Prosedur Pengujian	1. Pengguna membuka menu kelola status kehadiran. 2. Pengguna menekan tombol <i>edit</i> . 3. Pengguna mengubah formulir status absensi. 4. Pengguna menekan tombol simpan.
Masukan	1. Kode status absensi. 2. Nama status absensi. 3. Terhitung absen.
Hasil yang diharapkan	Status absensi tersunting dan notifikasi sukses muncul

Hasil pengujian	Berhasil
------------------------	----------

Tabel B.13.Detail Pengujian Fungsi Hapus Status Absensi

Nama Skenario Pengujian	Fungsionalitas mengelola status absensi
Kode	PF-012
Tujuan Pengujian	Menguji fungsi hapus status absensi.
Kondisi Awal	Pengguna merupakan administrator.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu kelola status kehadiran. 2. Pengguna menekan tombol hapus.
Masukan	
Hasil yang diharapkan	Status absensi terhapus dan notifikasi sukses muncul
Hasil pengujian	Berhasil

Tabel B.14.Detail Pengujian Fungsi Tambah Periode

Nama Skenario Pengujian	Fungsionalitas mengelola periode
Kode	PF-013
Tujuan Pengujian	Menguji fungsi tambah periode.
Kondisi Awal	<ol style="list-style-type: none"> 1. Pengguna merupakan administrator. 2. Belum ada periode aktif
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu kelola periode

	2. Pengguna menekan tombol tambah 3. Pengguna mengubah formulir tahun ajaran 4. Pengguna menekan tombol simpan
Masukan	1. Tahun ajaran. 2. Semester. 3. Tanggal akhir pembayaran. 4. Tanggal awal penyusunan KRS. 5. Tanggal akhir penyusunan KRS. 6. Tanggal akhir perubahan KRS. 7. Tanggal akhir pembatalan KRS. 8. Tanggal akhir penilaian.
Hasil yang diharapkan	Periode baru tersimpan dan notifikasi sukses muncul.
Hasil pengujian	Gagal. Error input tanggal

Tabel B.15.Detail Pengujian Fungsi Sunting Periode

Nama Skenario Pengujian	Fungsionalitas mengelola periode.
Kode	PF-014
Tujuan Pengujian	Menguji fungsi sunting periode.
Kondisi Awal	Pengguna merupakan administrator.
Prosedur Pengujian	1. Pengguna membuka menu kelola periode. 2. Pengguna menekan tombol <i>edit</i> . 3. Pengguna mengubah formulir periode. 4. Pengguna menekan tombol simpan.
Masukan	1. Tahun ajaran.

	2. Semester. 3. Tanggal akhir pembayaran. 4. Tanggal awal penyusunan KRS. 5. Tanggal akhir penyusunan KRS. 6. Tanggal akhir perubahan KRS. 7. Tanggal akhir pembatalan KRS. 8. Tanggal akhir penilaian.
Hasil yang diharapkan	Periode tersunting dan notifikasi sukses muncul
Hasil pengujian	Gagal. Error input tanggal

Tabel B.16.Detail Pengujian Fungsi Hapus Periode

Nama Skenario Pengujian	Fungsionalitas mengelola periode
Kode	PF-015
Tujuan Pengujian	Menguji fungsi hapus periode.
Kondisi Awal	Pengguna merupakan administrator.
Prosedur Pengujian	1. Pengguna membuka menu kelola periode. 2. Pengguna menekan tombol hapus.
Masukan	
Hasil yang diharapkan	Periode terhapus dan notifikasi sukses muncul
Hasil pengujian	Berhasil

Tabel B.17.Detail Pengujian Fungsi Tambah Pembelajaran

Nama Skenario Pengujian	Fungsionalitas mengelola pembelajaran
Kode	PF-016

Tujuan Pengujian	Menguji fungsi tambah pembelajaran.
Kondisi Awal	Pengguna merupakan tenaga kependidikan.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu kelola pembelajaran. 2. Pengguna menekan tombol tambah. 3. Pengguna mengubah formulir tahun ajaran. 4. Pengguna menekan tombol simpan.
Masukan	<ol style="list-style-type: none"> 1. Matakuliah. 2. Nama pembelajaran. 3. Kuota. 4. Pertemuan dalam seminggu. 5. Kelas untuk.
Hasil yang diharapkan	Pembelajaran baru tersimpan dan notifikasi sukses muncul.
Hasil pengujian	Berhasil

Tabel B.18.Detail Pengujian Fungsi Sunting Pembelajaran

Nama Skenario Pengujian	Fungsionalitas mengelola pembelajaran.
Kode	PF-017
Tujuan Pengujian	Menguji fungsi sunting pembelajaran.
Kondisi Awal	Pengguna merupakan tenaga kependidikan.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu kelola pembelajaran. 2. Pengguna menekan tombol <i>edit</i>.

	3. Pengguna mengubah formulir pembelajaran. 4. Pengguna menekan tombol simpan.
Masukan	1. Matakuliah. 2. Nama pembelajaran. 3. Kuota. 4. Pertemuan dalam seminggu. 5. Kelas untuk.
Hasil yang diharapkan	Pembelajaran tersunting dan notifikasi sukses muncul
Hasil pengujian	Berhasil

Tabel B.19.Detail Pengujian Fungsi Hapus Pembelajaran

Nama Skenario Pengujian	Fungsionalitas mengelola pembelajaran
Kode	PF-018
Tujuan Pengujian	Menguji fungsi hapus pembelajaran.
Kondisi Awal	Pengguna merupakan tenaga kependidikan.
Prosedur Pengujian	1. Pengguna membuka menu kelola pembelajaran. 2. Pengguna menekan tombol hapus
Masukan	
Hasil yang diharapkan	Pembelajaran terhapus dan notifikasi sukses muncul
Hasil pengujian	Berhasil

Tabel B.20.Detail Pengujian Fungsi Tambah Pengajar

Nama Skenario Pengujian	Fungsionalitas mengelola pembelajaran
Kode	PF-019
Tujuan Pengujian	Menguji fungsi tambah pengajar.
Kondisi Awal	Pengguna merupakan tenaga kependidikan.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu kelola pembelajaran. 2. Pengguna menekan tombol kelola pengajar. 3. Pengguna menekan tombol tambah pengajar. 4. Pengguna menekan tombol tambahkan.
Masukan	
Hasil yang diharapkan	Peengajar baru tertambahkan dan notifikasi sukses muncul.
Hasil pengujian	Berhasil

Tabel B.21.Detail Pengujian Fungsi Hapus Pengajar

Nama Skenario Pengujian	Fungsionalitas mengelola pembelajaran
Kode	PF-020
Tujuan Pengujian	Menguji fungsi hapus pengajar.
Kondisi Awal	Pengguna merupakan tenaga kependidikan.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu kelola pembelajaran. 2. Pengguna menekan tombol kelola pengajar.

	3. Pengguna menekan tombol hapus.
Masukan	
Hasil yang diharapkan	Peengajar terhapus dan notifikasi sukses muncul.
Hasil pengujian	Berhasil

Tabel B.22.Detail Pengujia Fungsi Tambah Peserta

Nama Skenario Pengujian	Fungsionalitas mengelola pembelajaran
Kode	PF-021
Tujuan Pengujian	Menguji fungsi tambah peserta.
Kondisi Awal	Pengguna merupakan tenaga kependidikan.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu kelola pembelajaran. 2. Pengguna menekan tombol peserta didik. 3. Pengguna menekan tombol tambah peserta. 4. Pengguna menekan tombol tambahkan.
Masukan	
Hasil yang diharapkan	Peserta baru tertambahkan dan notifikasi sukses muncul.
Hasil pengujian	Berhasil

Tabel B.23.Detail Pengujian Fungsi Tambah Peserta dari Rombel

Nama Skenario Pengujian	Fungsionalitas mengelola pembelajaran
Kode	PF-022

Tujuan Pengujian	Menguji fungsi tambah peserta dari rombel.
Kondisi Awal	Pengguna merupakan tenaga kependidikan.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu kelola pembelajaran. 2. Pengguna menekan tombol peserta didik. 3. Pengguna menekan tombol tambah dari rombel. 4. Pengguna menekan tombol tambahkan.
Masukan	
Hasil yang diharapkan	Anggota rombel menjadi peserta dan notifikasi sukses muncul.
Hasil pengujian	Berhasil

Tabel B.24.Detail Pegujian Fungsi Hapus Peserta

Nama Skenario Pengujian	Fungsionalitas mengelola pembelajaran
Kode	PF-023
Tujuan Pengujian	Menguji fungsi hapus peserta.
Kondisi Awal	Pengguna merupakan tenaga kependidikan.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu kelola pembelajaran. 2. Pengguna menekan tombol peserta didik. 3. Pengguna menekan tombol hapus
Masukan	

Hasil yang diharapkan	Peserta terhapus dan notifikasi sukses muncul.
Hasil pengujian	Berhasil

Tabel B.25.Detail Pengujian Fungsi Tambah Rombongan Belajar

Nama Skenario Pengujian	Fungsionalitas mengelola rombongan belajar
Kode	PF-024
Tujuan Pengujian	Menguji fungsi tambah rombongan belajar.
Kondisi Awal	Pengguna merupakan tenaga kependidikan.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu kelola rombongan belajar. 2. Pengguna menekan tombol tambah. 3. Pengguna mengubah formulir tahun ajaran. 4. Pengguna menekan tombol simpan.
Masukan	Nama rombongan belajar.
Hasil yang diharapkan	Rombongan belajar baru tersimpan dan notifikasi sukses muncul.
Hasil pengujian	Berhasil

Tabel B.26.Detail Pengujian Fungsi Sunting Rombongan Belajar

Nama Skenario Pengujian	Fungsionalitas mengelola rombongan belajar.
Kode	PF-025

Tujuan Pengujian	Menguji fungsi sunting rombongan belajar.
Kondisi Awal	Pengguna merupakan tenaga kependidikan.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu kelol rombongan belajar. 2. Pengguna menekan tombol <i>edit</i>. 3. Pengguna mengubah formulir rombongan belajar. 4. Pengguna menekan tombol simpan.
Masukan	Nama rombongan belajar.
Hasil yang diharapkan	Rombongan belajar tersunting dan notifikasi sukses muncul
Hasil pengujian	Berhasil

Tabel B.27.Detail Pengujian Fungsi Hapus Rombel

Nama Skenario Pengujian	Fungsionalitas mengelola rombongan belajar
Kode	PF-026
Tujuan Pengujian	Menguji fungsi hapus rombongan belajar.
Kondisi Awal	Pengguna merupakan tenaga kependidikan.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu kelola rombongan belajar. 2. Pengguna menekan tombol hapus.
Masukan	

Hasil yang diharapkan	Rombongan belajar terhapus dan notifikasi sukses muncul
Hasil pengujian	Berhasil

Tabel B.28.Detail Pengujian Fungsi Tambah Anggota

Nama Skenario Pengujian	Fungsionalitas mengelola rombongan belajar
Kode	PF-027
Tujuan Pengujian	Menguji fungsi tambah anggota.
Kondisi Awal	Pengguna merupakan tenaga kependidikan.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu kelola rombongan belajar. 2. Pengguna menekan tombol isi rombongan belajar. 3. Pengguna menekan tombol tambah. 4. Pengguna menekan tombol tambahkan.
Masukan	
Hasil yang diharapkan	Peserta baru tertambahkan dan notifikasi sukses muncul.
Hasil pengujian	Berhasil

Tabel B.29.Detail Pengujian Fungsi Hapus Anggota

Nama Skenario Pengujian	Fungsionalitas mengelola rombongan belajar
Kode	PF-028
Tujuan Pengujian	Menguji fungsi hapus anggota.
Kondisi Awal	Pengguna merupakan tenaga kependidikan.

Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu kelola rombongan belajar. 2. Pengguna menekan tombol isi rombongan belajar. 3. Pengguna menekan tombol hapus
Masukan	
Hasil yang diharapkan	Peserta terhapus dan notifikasi sukses muncul.
Hasil pengujian	Berhasil

Tabel B.30.Detail Pengujian Fungsi Tambah Anak Wali

Nama Skenario Pengujian	Fungsionalitas mengelola pendamping akademik
Kode	PF-029
Tujuan Pengujian	Menguji fungsi tambah anak wali.
Kondisi Awal	Pengguna merupakan tenaga kependidikan.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu kelola pendamping akademik. 2. Pengguna menekan tombol kelola perwalian. 3. Pengguna menekan tombol tambah anak wali. 4. Pengguna menekan tombol tambahkan
Masukan	
Hasil yang diharapkan	Anak wali baru tertambahkan dan notifikasi sukses muncul.
Hasil pengujian	Berhasil

Nama Skenario Pengujian	Fungsionalitas mengelola pendamping akademik
Kode	PF-030
Tujuan Pengujian	Menguji fungsi lepasanak wali.
Kondisi Awal	Pengguna merupakan tenaga kependidikan.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu kelola pendamping akademik. 2. Pengguna menekan tombol kelola perwalian. 3. Pengguna menekan tombol lepas
Masukan	
Hasil yang diharapkan	Anak wali terhapus dan notifikasi sukses muncul.
Hasil pengujian	Berhasil

Tabel B.31.Detail Pengujian Fungsi Absensi Peserta Didik

Nama Skenario Pengujian	Fungsionalitas mengelola absensi
Kode	PF-031
Tujuan Pengujian	Menguji fungsi absensi peserta didik.
Kondisi Awal	Pengguna merupakan tenaga kependidikan.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu absensi. 2. Pengguna menekan tombol absensi peserta didik

	3. Pengguna masukan absensi peserta didik. 4. Pengguna menekan tombol simpan.
Masukan	Absensi peserta didik tiap pertemuan
Hasil yang diharapkan	Absensi tersimpan dan notifikasi sukses muncul.
Hasil pengujian	Berhasil

Tabel B.32.Detail Pengujian Fungsi Absensi Pendidik

Nama Skenario Pengujian	Fungsionalitas mengelola absensi
Kode	PF-032
Tujuan Pengujian	Menguji fungsi absensi pendidik.
Kondisi Awal	Pengguna merupakan tenaga kependidikan.
Prosedur Pengujian	1. Pengguna membuka menu absensi. 2. Pengguna menekan tombol absensi pendidik 3. Pengguna masukan absensi pendidik. 4. Pengguna menekan tombol simpan.
Masukan	Absensi pendidik tiap pertemuan
Hasil yang diharapkan	Absensi tersimpan dan notifikasi sukses muncul.
Hasil pengujian	Berhasil

Tabel B.33.Detail Pengujian Fungsi Mengambil Pembelajaran

Nama Skenario Pengujian	Fungsionalitas menyusun KRS
Kode	PF-033
Tujuan Pengujian	Menguji fungsi mengambil pembelajaran.
Kondisi Awal	<ol style="list-style-type: none"> 1. Pengguna merupakan peserta didik. 2. Masih dalam masa penyusunan KRS. 3. KRS belum disetujui. 4. Sisa pengambilan KRS masih ada. 5. Peserta dalam pembelajaran belum memenuhi kuota.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu kartu rencana studi. 2. Pengguna memilih pembelajaran. 3. Pengguna menekan tombol ambil. 4. Pengguna menekan tombol simpan.
Masukan	
Hasil yang diharapkan	Pembelajaran terambil dan notifikasi sukses muncul.
Hasil pengujian	Berhasil

Tabel B.34.Detail Pengujian Fungsi Menyusun KRS

Nama Skenario Pengujian	Fungsionalitas menyusun KRS
Kode	PF-034
Tujuan Pengujian	Menguji fungsi menghapus pembelajaran.

Kondisi Awal	1. Pengguna merupakan peserta didik. 2. Masih dalam masa penyusunan KRS. 3. KRS belum di setujui.
Prosedur Pengujian	1. Pengguna membuka menu kertu rencana studi. 2. Pengguna menekan tombol hapus.
Masukan	
Hasil yang diharapkan	Pembelajaran terhapus dan notifikasi sukses muncul.
Hasil pengujian	Berhasil

Tabel B.35.Detail Pengujian Fungsi Menambah Pembelajaran

Nama Skenario Pengujian	Fungsionalitas mencetak KRS
Kode	PF-035
Tujuan Pengujian	Menguji fungsi menambah pembelajaran.
Kondisi Awal	Pengguna merupakan peserta didik.
Prosedur Pengujian	1. Pengguna membuka menu kertu rencana studi. 2. Pengguna menekan tombol cetak KRS
Masukan	
Hasil yang diharapkan	KRS versi cetak muncul.

Hasil pengujian	Berhasil
------------------------	----------

Tabel B.36.Detail Pengujian Fungsi Rekap Absensi Peserta Didik

Nama Skenario Pengujian	Fungsionalitas laporan pertemuan
Kode	Menguji fungsi rekap absensi peserta didik.
Tujuan Pengujian	PF-036
Kondisi Awal	Pengguna merupakan kepala satuan manajemen.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu laporan pertemuan. 2. Pengguna menekan tombol absensi peserta didik.
Masukan	
Hasil yang diharapkan	Muncul rekap absensi peserta didik.
Hasil pengujian	Berhasil

Tabel B.37.Detail Pengujian Fungsi Rekap Absensi Pendidik

Nama Skenario Pengujian	Fungsionalitas laporan pertemuan
Kode	PF-037
Tujuan Pengujian	Menguji fungsi rekap absensi pendidik.
Kondisi Awal	Pengguna merupakan kepala satuan manajemen.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu laporan pertemuan.

	2. Pengguna menekan tombol absensi pendidik.
Masukan	
Hasil yang diharapkan	Muncul rekap absensi pendidik.
Hasil pengujian	Berhasil

Tabel B.38.Detail Pengujian Fungsi Rekap Berita Acara

Nama Skenario Pengujian	Fungsionalitas laporan pertemuan
Kode	PF-038
Tujuan Pengujian	Menguji fungsi rekap berita acara.
Kondisi Awal	Pengguna merupakan kepala satuan manajemen.
Prosedur Pengujian	1. Pengguna membuka menu laporan pertemuan. 2. Pengguna menekan tombol berita acara
Masukan	
Hasil yang diharapkan	Muncul rekap berita acara.
Hasil pengujian	Berhasil

Tabel B.39.Detail Pengujian Fungsi Persetujuan KRS

Nama Skenario Pengujian	Fungsionalitas menyetujui KRS
Kode	PF-039
Tujuan Pengujian	Menguji fungsi persetujuan RKS.
Kondisi Awal	

	<ol style="list-style-type: none"> 1. Pengguna merupakan pendidik. 2. Masih dalam masa perubahan KRS. 3. KRS belum disetujui.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu persetujuankurtu rencana studi. 2. Pengguna memilih peserta didik. 3. Pengguna menekan tombol setuju.
Masukan	
Hasil yang diharapkan	KRS tersetujui dan notifikasi sukses muncul.
Hasil pengujian	Berhasil

Tabel B.40.Detail Pengujian Fungsi Membatalkan Persetujuan

Nama Skenario Pengujian	Fungsionalitas menyetujui KRS
Kode	PF-040
Tujuan Pengujian	Menguji fungsi membatalkan persetujuan.
Kondisi Awal	<ol style="list-style-type: none"> 1. Pengguna merupakan pendidik. 2. Masih dalam masa perubahan KRS. 3. KRS belum disetujui.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu kurtu rencana studi. 2. Pengguna menekan tombol batalkan persetujuan.
Masukan	

Hasil yang diharapkan	Status KRS belum disetujui dan notifikasi sukses muncul.
Hasil pengujian	Berhasil

Tabel B.41.Detail Pengujian Fungsi Berita Acara

Nama Skenario Pengujian	Fungsionalitas mengelola berita acara
Kode	PF-041
Tujuan Pengujian	Menguji fungsi tambah berita acara.
Kondisi Awal	Pengguna merupakan pengajar.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu berita acara. 2. Pengguna menekan tombol lihat berita acara. 3. Pengguna menekantombolisi berita acara.
Masukan	<ol style="list-style-type: none"> 1. Tanggal Pertemuan. 2. Materi. 3. Kendala pertemuan. 4. Tanggapan peserta didik.
Hasil yang diharapkan	Berita acara baru tersimpan dan notifikasi sukses muncul.
Hasil pengujian	Gagal. Error input tanggal

Tabel B.42.Detail Pengujian Fungsi Pembayaran Peserta Didik

Nama Skenario Pengujian	Fungsionalitas laporan pembayaran
Kode	PF-042

Tujuan Pengujian	Menguji fungsi rekap pembayaran peserta didik.
Kondisi Awal	Pengguna merupakan kepala satuan manajemen.
Prosedur Pengujian	Pengguna membuka menu laporan pembayaran.
Masukan	
Hasil yang diharapkan	Muncul rekap pembayaran peserta didik.
Hasil pengujian	Berhasil

Tabel B.43.Detail Pengujian Fungsi Sungting Berita Acara

Nama Skenario Pengujian	Fungsionalitas mengelola berita acara
Kode	PF-043
Tujuan Pengujian	Menguji fungsi sunting berita acara.
Kondisi Awal	Pengguna merupakan pengajar.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu berita acara. 2. Pengguna menekan tombol lihat berita acara. 3. Pengguna menekantombol <i>edit</i> berita acara.
Masukan	<ol style="list-style-type: none"> 1. Tanggal Pertemuan. 2. Materi. 3. Kendala pertemuan. 4. Tanggapan peserta didik.
Hasil yang diharapkan	Berita acara tersunting dan notifikasi sukses muncul.
Hasil pengujian	Gagal. Error input tanggal

Tabel B.44.Detail Pengujian Fungsi Pengambilan Mata Kuliah ITS

Nama Skenario Pengujian	Pengaturan pengambilan matakuliah ITS
Kode	PF-044
Tujuan Pengujian	Menguji aturan pengambilan matakulliah ITS
Kondisi Awal	Pengguna merupakan administrator.
Prosedur Pengujian	1. Pengguna membuka menu kelola periode. 2. Pengguna menyunting periode aktif.
Masukan	Tanggal pembayaran < Tanggal awal penyusunan
Hasil yang diharapkan	Peserta didik dapat mengambil matakuliah jika telah melakukan pembayaran
Hasil pengujian	Berhasil

Tabel B.45.Detail Pengujian Fungsi Pengambilan Mata Kuliah PENS

Nama Skenario Pengujian	Pengaturan pengambilan matakuliah PENS
Kode	PF-045
Tujuan Pengujian	Menguji aturan pengambilan matakulliah PENS
Kondisi Awal	Pengguna merupakan administrator.
Prosedur Pengujian	1. Pengguna membuka menu kelola periode. 2. Pengguna menyunting periode aktif.

Masukan	Tidak memperbolehkan penyusunan KRS
Hasil yang diharapkan	Peserta didik tidak dapat mengambil matakuliah. Masa penyusunan tidak ada.
Hasil pengujian	Berhasil

Tabel B.46.Detail Pengujian Fungsi Pengambilan Matakuliah UPN

Nama Skenario Pengujian	Pengaturan pengambilan matakuliah UPN
Kode	PF-047
Tujuan Pengujian	Menguji autan pengambilan matakulliah UPN
Kondisi Awal	Pengguna merupakan administrator.
Prosedur Pengujian	1. Pengguna membuka menu kelola periode. 2. Pengguna menyunting periode aktif.
Masukan	Tanggal pembayaran > Tanggal akhir penyusunan
Hasil yang diharapkan	Peserta didik dapat mengambil matakuliah walaupun belum melakukan pembayaran
Hasil pengujian	Berhasil

Tabel B.47.Detail Pengujian Fungsi Perubahan Kebijakan

Nama Skenario Pengujian	Perubahan batas kehadiran minimum peserta didik
Kode	PF-046

Tujuan Pengujian	Menguji perubahan kebijakan.
Kondisi Awal	Pengguna merupakan administrator.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu kelola tahun ajaran. 2. Pengguna menyunting tauh ajaran.
Masukan	Minimal pertemuan pembelajaran: 90%
Hasil yang diharapkan	Keterangan dalam rekap absensi berubah.
Hasil pengujian	Berhasil

LAMPIRAN C

Tabel C.1.Detail Perbaikan Fungsi Akses Menu

Kode	PF-000
Tujuan Pengujian	Uji coba akses menu.
Pesan Kesalahan	Servlet Error() 500
Perubahan	File: sia-service/src/main/reources/META-INF/spring/sia-service.xml Perubahan: <ol style="list-style-type: none"> (Line 100) perubahan directory penyimpanan modul
	File: sia-web/src/main/webapp/WEB-INF/spring-beans/modul/sia-modul-servlet.xml Perubahan: <ol style="list-style-type: none"> (Line 100) perubahan directory penyimpanan modul
Hasil Pengujian Ulang	Berhasil

Tabel C.2. Detail Perbaikan Fungsi Tambah Periode Baru

Kode	PF-013
Tujuan Pengujian	Tambah periode baru.
Pesan Kesalahan	Gagal input tanggal
Perubahan	File: sia-modul-domain/src/main/java/com/sia/modul/domain/TglSmt.java Perubahan: <ol style="list-style-type: none"> (Line 37,42, 48, 54, 60, 66) Hapus anotasi type tanggal joda date time.

Hasil Pengujian Ulang	Berhasil
------------------------------	----------

Tabel C.3.Detail Perbaikan Fungsi Edit Periode

Kode	PF-014
Tujuan Pengujian	Edit periode
Pesan Kesalahan	Gagal input tanggal
Perubahan	File: sia-modul-domain/src/main/java/com/sia/modul/domain/TglSmt.java Perubahan: 2. (Line 37,42, 48, 54, 60, 66) Hapus anotasi type tanggal joda date time.
Hasil Pengujian Ulang	Berhasil

Tabel C.4.Detail Perbaikan Fungsi Tambah Berita Acara

Kode	PF -041
Tujuan Pengujian	Tambah berita acara
Pesan Kesalahan	Gagal input tanggal
Perubahan	File: sia-modul-domain/src/main/java/com/sia/modul/domain/TglSmt.java Perubahan: 3. (Line 37,42, 48, 54, 60, 66) Hapus anotasi type tanggal joda date time.
Hasil Pengujian Ulang	Berhasil

Tabel C.5.Detail Perbaikan Fungsi Edit Berita Acara

Kode	PF -043
Tujuan Pengujian	Edit berita acara
Pesan Kesalahan	Gagal input tanggal
Perubahan	File: sia-modul-domain/src/main/java/com/sia/modul/domain/TglSmt.java Perubahan: 4. (Line 37,42, 48, 54, 60, 66) Hapus anotasi type tanggal joda date time.
Hasil Pengujian Ulang	Berhasil

LAMPIRAN D

Tabel D.6. Hasil Pengukuran Kualitas pada *Service Layer*

Aspek	Deskripsi	
coupling of component conformance (mmo-1-G)	mengukur tingkat independensi suatu komponen pada program	
Sub-karakteristik	Modularity	
A	a = Jumlah komponen yang diimplementasikan dengan dampak minimal pada komponen lainnya	16
B	b = jumlah komponen yang harus independent	32
Hasil	0.5	

Tabel D.7. Hasil Pengukuran Kualitas Cyclomatic Complexity pada *Service Layer*

Aspek	Deskripsi	
cyclomatic complexity (MMo-2-S)	Mengukur tingkat cyclomatic dari program	
Sub-karakteristik	Modularity	
A	jumlah kelas yang memiliki kompleksitas cyclomatic melebihi trashhold	6
B	jumlah kelas.	723
Hasil	0.991701245	

Tabel D.8. Hasil Pengukuran Kualitas Reusability of Assets pada *Service Layer*

Aspek	Deskripsi	
reusability of assets (MRe-1-G)	mengukur tingkat penggunaan ulang komponen	
Sub-karakteristik	Reusability	
A	jumlah komponen dari sistem yang dapat digunakan ulang	70
B	jumlah komponen dari sistem	86
Hasil	0.813953488	

Tabel D.9.Hasil Pengukuran Kualitas Conformance to Coding Rule pada *Service Layer*

Aspek	Deskripsi	
conformance to coding rule (MRe-2-S)	mengukur tingkat konsistensi penggunaan aturan pemrograman	
Sub-karakteristik	Reusability	
A	jumlah komponen yang menerapkan aturan pemrograman	84
B	jumlah komponen	147
Hasil	0.571428571	

Tabel D.10.Hasil Pengukuran Kualitas System Log Completeness Conformance pada *Service Layer*

Aspek	Deskripsi	
system log completeness conformance (MAN-1-G)	system log completeness conformance	
Sub-karakteristik	Analysability	
A	jumlah log yang terekam	5

B	jumlah log yang harusnya ada untuk audit sistem	15
Hasil	0.33333333	

Tabel D.11. Hasil Pengukuran Kualitas Diagnosis Function Effectiveness pada *Service Layer*

Aspek	Deskripsi	
diagnosis function effectiveness (MAN-2-S)	Mengukur sejauh mana efektifitas implementasi fitur yang sesuai dengan requirement dibandingkan dengan fitur yang telah diimplementasikan	
Sub-karakteristik	Analysability	
A	a = jumlah fitur yang menjawab requirement	15
B	b = jumlah fitur yang ada pada program	15
Hasil	1	

Tabel D.12. Hasil Pengukuran Kualitas Diagnosis Function Suffuance Conformance pada *Service Layer*

Aspek	Deskripsi	
diagnosis function suffuance conformance (MAN-3-S)	mengukur sejauh mana fitur-fitur memenuhi spesifikasi kebutuhan.	
Sub-karakteristik	Analysability	
A	a = jumlah fitur yang telah dibuat berdasarkan requirement	15
B	b = jumlah fitur yang seharusnya dibuat sesuai dengan requirement.	15
Hasil	1	

Tabel D.13.Hasil Pengukuran Kualitas Test function complateness conformance pada *Service Layer*

Aspek	Deskripsi	
Test function complateness conformance (MTg-1-G)	mengukur kelengkapan uji coba terhadap sistem	
Sub-karakteristik	Testability	
A	jumlah uji coba yang dilakukan	50
B	jumlah uji coba yang harusnya dilakukan	50
Hasil	1	

Tabel D.14.Hasil Pengukuran Kualitas Autonomus testability pada *Service Layer*

Aspek	Deskripsi	
Autonomus testability (MTg-2-S)	mengukur kelengkapan uji coba terhadap sistem	
Sub-karakteristik	Testability	
A	jumlah stub yang dapat berjalan pada dependency test	0
B	jumlah uji coba yang harusnya dilakukan	50
Hasil	0	

Tabel D.15.Hasil Pengukuran Kualitas Kemampuan Restart pada *Service Layer*

Aspek	Deskripsi	
kemampuan restart (MTg-3-S)	mengukur kemudahan uji coba ulang setelah melakukan perbaikan sistem	
Sub-karakteristik	Testability	

A	jumlah uji coba yang memiliki titik pause dan restart	0
B	jumlah uji coba yang memiliki titik pause	50
Hasil	0	

Tabel D.16.Hasil Pengukuran Kualitas pada Coupling of component conformance *Transaction Script*

Aspek	Deskripsi	
coupling of component conformance (mmo-1-G)	mengukur tingkat independensi suatu komponen pada program	
Sub-karakteristik	Modularity	
A	a = Jumlah komponen yang diimplementasikan dengan dampak minimal pada komponen lainnya	16
B	b = jumlah komponen yang harus independent	25
Hasil	0.64	

Tabel D.17.Hasil Pengukuran Kualitas Cyclomatic Complexity pada *Transaction Script*

Aspek	Deskripsi	
cyclomatic complexity (MMo-2-S)	Mengukur tingkat cyclomatic dari program	
Sub-karakteristik	Modularity	
A	jumlah kelas yang memiliki kompleksitas cyclomatic melebihi trashhold	6
B	jumlah kelas.	454
Hasil	0.986784141	

**Tabel D.18.Hasil Pengukuran Kualitas Reusability of Assets
pada *Transaction Script***

Aspek	Deskripsi	
reusability of assets (MRe-1-G)	mengukur tingkat penggunaan ulang komponen	
Sub-karakteristik	Reusability	
A	jumlah komponen dari sistem yang dapat digunakan ulang	9
B	jumlah komponen dari sistem	41
Hasil	0.219512195	

**Tabel D.19.Hasil Pengukuran Kualitas Conformance to
Coding Rule pada *Transaction Script***

Aspek	Deskripsi	
conformance to coding rule (MRe-2-S)	mengukur tingkat konsistensi penggunaan aturan pemrograman	
Sub-karakteristik	Reusability	
A	jumlah komponen yang menerapkan aturan pemrograman	8
B	jumlah komponen	57
Hasil	0.140350877	

Tabel D.20. Hasil Pengukuran Kualitas System Log Completeness Conformance pada *Transaction Script*

Aspek	Deskripsi	
system log completeness conformance (MAN-1-G)	system log completeness conformance	
Sub-karakteristik	Analysability	
A	jumlah log yang terekam	15
B	jumlah log yang harusnya ada untuk audit sistem	15
Hasil	1	

Tabel D.21. Hasil Pengukuran Kualitas Diagnosis Function Effectiveness pada *Transaction Script*

Aspek	Deskripsi	
diagnosis function effectiveness (MAN-2-S)	Mengukur sejauh mana efektifitas implementasi fitur yang sesuai dengan requirement dibandingkan dengan fitur yang telah diimplementasikan	
Sub-karakteristik	Analysability	
A	a = jumlah fitur yang menjawab requirement	15
B	b = jumlah fitur yang ada pada program	15
Hasil	1	

Tabel D.22. Hasil Pengukuran Kualitas Diagnosis Function Suffiance Conformance pada *Transaction Script*

Aspek	Deskripsi	
diagnosis function suffiance conformance (MAN-3-S)	mengukur sejauh mana fitur-fitur memenuhi spesifikasi kebutuhan.	

Sub-karakteristik	Analysability	
A	a = jumlah fitur yang telah dibuat berdasarkan requirement	15
B	b = jumlah fitur yang seharusnya dibuat sesuai dengan requirement.	15
Hasil	1	

Tabel D.23. Hasil Pengukuran Kualitas Test function complateness conformance pada *Transaction Script*

Aspek	Deskripsi	
Test function complateness conformance (MTg-1-G)	mengukur kelengkapan uji coba terhadap sistem	
Sub-karakteristik	Testability	
A	jumlah uji coba yang dilakukan	50
B	jumlah uji coba yang harusnya dilakukan	50
Hasil	1	

Tabel D.24. Hasil Pengukuran Kualitas Autonomus testability pada *Transaction Script*

Aspek	Deskripsi	
Autonomus testability (MTg-2-S)	mengukur kelengkapan uji coba terhadap sistem	
Sub-karakteristik	Testability	
A	jumlah stub yang dapat berjalan pada dependency test	0
B	jumlah uji coba yang harusnya dilakukan	50

Hasil	0
--------------	---

Tabel D.25. Hasil Pengukuran Kualitas Kemampuan Restart pada *Transaction Script*

Aspek	Deskripsi	
kemampuan restart (MTg-3-S)	mengukur kemudahan uji coba ulang setelah melakukan perbaikan sistem	
Sub-karakteristik	Testability	
A	jumlah uji coba yang memiliki titik pause dan restart	0
B	jumlah uji coba yang memiliki titik pause	50
Hasil	0	

Tabel D.26. Hasil Pengukuran Kualitas Coupling of component conformance pada *Domain Model*

Aspek	Deskripsi	
coupling of component conformance (MMo-2-S)	mengukur tingkat independensi suatu komponen pada program	
Sub-karakteristik	Modularity	
A	a = Jumlah komponen yang diimplementasikan dengan dampak minimal pada komponen lainnya	16
B	b = jumlah komponen yang harus independent	40
Hasil	0.4	

Tabel D.27.Hasil Pengukuran Kualitas Cyclomatic Complexity pada *Domain Model*

Aspek	Deskripsi	
cyclomatic complexity (MMo-2-S)	Mengukur tingkat cyclomatic dari program	
Sub-karakteristik	Modularity	
A	jumlah kelas yang memiliki kompleksitas cyclomatic melebihi trashhold	6
B	jumlah kelas.	576
Hasil	0.989583333	

Tabel D.28.Hasil Pengukuran Kualitas Reusability of Assets pada *Domain Model*

Aspek	Deskripsi	
reusability of assets (MRe-1-G)	mengukur tingkat penggunaan ulang komponen	
Sub-karakteristik	Reusability	
A	jumlah komponen dari sistem yang dapat digunakan ulang	40
B	jumlah komponen dari sistem	56
Hasil	0.714285714	

Tabel D.29.Hasil Pengukuran Kualitas Conformance to Coding Rule pada *Domain Model*

Aspek	Deskripsi	
conformance to coding rule (MRe-2-S)	mengukur tingkat konsistensi penggunaan aturan pemrograman	
Sub-karakteristik	Reusability	

A	jumlah komponen yang menerapkan aturan pemrograman	26
B	jumlah komponen	87
Hasil	0.298850575	

Tabel D.30.Hasil Pengukuran Kualitas System Log Completeness Conformance pada Domain Model

Aspek	Deskripsi	
system log completeness conformance (MAN-1-G)	system log completeness conformace	
Sub-karakteristik	Analysability	
A	jumlah log yang terekam	15
B	jumlah log yang harusnya ada untuk audit sistem	15
Hasil	1	

Tabel D.31.Hasil Pengukuran Kualitas Diagnosis Function Effectiveness pada Domain Model

Aspek	Deskripsi	
diagnosis function effectiveness (MAN-2-S)	Mengukur sejauh mana efektifitas implementasi fitur yang sesuai dengan requirement dibandingkan dengan fitur yang telah diimplementasikan	
Sub-karakteristik	Analysability	
A	a = jumlah fitur yang menjawab requerement	15
B	b = jumlah fitur yang ada pada program	15
Hasil	1	

Tabel D.32. Hasil Pengukuran Kualitas Diagnosis Function Suffience Conformance pada *Domain Model*

Aspek	Deskripsi	
diagnosis function suffiance conformance (MAN-3-S)	mengukur sejauh mana fitur-fitur memenuhi spesifikasi kebutuhan.	
Sub-karakteristik	Analysability	
A	a = jumlah fitur yang telah dibuat berdasarkan requirement	15
B	b = jumlah fitur yang seharusnya dibuat sesuai dengan requirement.	15
Hasil	1	

Tabel D.33. Hasil Pengukuran Kualitas Test function complateness conformance pada *Domain Model*

Aspek	Deskripsi	
Test function complateness conformance (MTg-1-G)	mengukur kelengkapan uji coba terhadap sistem	
Sub-karakteristik	Testability	
A	jumlah uji coba yang dilakukan	50
B	jumlah uji coba yang harusnya dilakukan	50
Hasil	1	

Tabel D.34. Hasil Pengukuran Kualitas Autonomus testability pada *Domain Model*

Aspek	Deskripsi	
Autonomus testability (MTg-2-S)	mengukur kelengkapan uji coba terhadap sistem	
Sub-karakteristik	Testability	
A	jumlah stub yang dapat berjalan pada dependency test	0
B	jumlah uji coba yang harusnya dilakukan	50
Hasil	0	

Tabel D.35. Hasil Pengukuran Kualitas Kemampuan Restart pada *Domain Model*

Aspek	Deskripsi	
kemampuan restart (MTg-3-S)	mengukur kemudahan uji coba ulang setelah melakukan perbaikan sistem	
Sub-karakteristik	Testability	
A	jumlah uji coba yang memiliki titik pause dan restart	0
B	jumlah uji coba yang memiliki titik pause	50
Hasil	0	

Tabel D.36. Hasil Pengukuran Kualitas Coupling of component conformance pada *Table Module*

Aspek	Deskripsi
coupling of component conformance (MMo-1-G)	mengukur tingkat independensi suatu komponen pada program
Sub-karakteristik	Modularity

A	a = Jumlah komponen yang diimplementasikan dengan dampak minimal pada komponen lainnya	16
B	b = jumlah komponen yang harus independent	39
Hasil	0.41025641	

Tabel D.37. Hasil Pengukuran Kualitas cyclomatic complexity pada *Table Module*

Aspek	Deskripsi	
cyclomatic complexity (MMo-2-S)	Mengukur tingkat cyclomatic dari program	
Sub-karakteristik	Modularity	
A	jumlah kelas yang memiliki kompleksitas cyclomatic melebihi trashhold	6
B	jumlah kelas.	450
Hasil	0.986666667	

Tabel D.38. Hasil Pengukuran Kualitas reusability of assets pada *Table Module*

Aspek	Deskripsi	
reusability of assets (MRe-1-G)	mengukur tingkat penggunaan ulang komponen	
Sub-karakteristik	Reusability	
A	jumlah komponen dari sistem yang dapat digunakan ulang	39
B	jumlah komponen dari sistem	55
Hasil	0.709090909	

Tabel D.39. Hasil Pengukuran Kualitas conformance to coding rule pada *Table Module*

Aspek	Deskripsi	
conformance to coding rule (MRe-2-S)	mengukur tingkat konsistensi penggunaan aturan pemrograman	
Sub-karakteristik	Reusability	
A	jumlah komponen yang menerapkan aturan pemrograman	34
B	jumlah komponen	85
Hasil	0.4	

Tabel D.40. Hasil Pengukuran Kualitas system log completeness conformance pada *Table Module*

Aspek	Deskripsi	
system log completeness conformance (MAN-1-G)	system log completeness conformace	
Sub-karakteristik	Analysability	
A	jumlah log yang terekam	15
B	jumlah log yang harusnya ada untuk audit sistem	15
Hasil	1	

Tabel D.41. Hasil Pengukuran Kualitas diagnosis function effectiveness pada *Table Module*

Aspek	Deskripsi	
diagnosis function effectiveness (MAN-2-S)	Mengukur sejauh mana efektifitas implementasi fitur yang sesuai dengan requirement dibandingkan dengan fitur yang telah diimplementasikan	
Sub-karakteristik	Analysability	
A	a = jumlah fitur yang menjawab requirement	15
B	b = jumlah fitur yang ada pada program	15
Hasil	1	

Tabel D.42. Hasil Pengukuran Kualitas diagnosis function suffiance conformance pada *Table Module*

Aspek	Deskripsi	
diagnosis function suffiance conformance (MAN-3-S)	mengukur sejauh mana fitur-fitur memenuhi spesifikasi kebutuhan.	
Sub-karakteristik	Analysability	
A	a = jumlah fitur yang telah dibuat berdasarkan requirement	15
B	b = jumlah fitur yang seharusnya dibuat sesuai dengan requirement.	15
Hasil	1	

Tabel D.43. Hasil Pengukuran Kualitas Test function complateness conformance pada *Table Module*

Aspek	Deskripsi
Test function complateness	mengukur kelengkapan uji coba terhadap sistem

conformance (MTg-1-G)		
Sub-karakteristik	Testability	
A	jumlah uji coba yang dilakukan	50
B	jumlah uji coba yang harusnya dilakukan	50
Hasil	1	

Tabel D.44. Hasil Pengukuran Kualitas Autonomus testability pada *Table Module*

Aspek	Deskripsi	
Autonomus testability (MTg-2-S)	mengukur kelengkapan uji coba terhadap sistem	
Sub-karakteristik	Testability	
A	jumlah stub yang dapat berjalan pada dependency test	0
B	jumlah uji coba yang harusnya dilakukan	50
Hasil	0	

Tabel D.45. Hasil Pengukuran Kualitas Kemampuan Restart pada *Table Module*

Aspek	Deskripsi	
kemampuan restart (MTg-3-S)	mengukur kemudahan uji coba ulang setelah melakukan perbaikan sistem	
Sub-karakteristik	Testability	
A	jumlah uji coba yang memiliki titik pause dan restart	0
B	jumlah uji coba yang memiliki titik pause	50
Hasil	0	

BIODATA PENULIS

Prasetya Gilang Nuswantara adalah Mahasiswa Teknik Informatika ITS angkatan 2013. Pria yang akrab dipanggil Gilang ini lahir Di Kediri, 29 Oktober 1994. Dalam perjalanan hidupnya hingga saat ini Gilang



lahir dan besar di Kota Kediri. Dia juga menjalani sekolah dasar hingga SMA di Kediri. Riwayat pendidikan dasar hingga menengahnya adalah sebagai berikut SDN Banjaran 4 Kediri (lulus 2007), SMPN 1 Kediri (lulus 2010) dan SMAN 2 Kediri (lulus 2013).

Masa kuliah penulis isi dengan mengikuti kegiatan perkuliahan, organisasi dan kompetisi. Pengalaman organisasi yang dimiliki adalah staff Ristek HMTC, staff Ristek BEMf, staff keilmuan KMI, Koordinator Kompetisi Robot ITS Expo dan Kadep Ristek HMTC. Saat ini pun gilang sedang aktif sebagai Kepala Divisi Eksternal Trainer Keilmiah ITS dan Kepala divisi Research dan Development Administrator Laboratorium RPL. Sedangkan kompetisi penulis telah mengikuti beberapa kompetisi seperti Gemastik, PKM, Compfest dan Lomba-lomba tentang informatika lainnya. Penulis juga pernah menjadi finalis di lomba GIS Nasional yang diadakan oleh UGM serta menjadi juara 2 pada lomba Developer Competition Nasional yang diadakan oleh FTIf ITS.

Selain sebagai mahasiswa, penulis juga merupakan seorang pengembang Teknologi Informasi khususnya pada bidang Software Engineer. Hal ini terbukti dengan pengambilan bidang minat Software Engineer di tahun terakhirnya. Selain itu juga Sekarang Gilang sedang bekerja sebagai Software Engineer di Software House Artcak Studio dan telah mengerjakan beberapa project disana. Selain itu penulis juga memiliki startup dibidang pendidikan yang bernama AkuPintar dan penulis berposisi sebagai CTO.